

# Distribuert samarbeidende organisering av musikkmateriale

*Design og evaluering av «Collaborative Music Organizer»*

Av

Bjarte Konglevoll Helland



Institutt for informasjons- og medievitenskap,

Universitetet i Bergen

Våren 2008

Fullført som del av kravet til graden

«Master i informasjonsvitenskap»

<b>1. INTRODUKSJON .....</b>	<b>1</b>
1.1 Studiets struktur .....	2
<b>2. BAKGRUNN.....</b>	<b>4</b>
2.1 Begrepsforklaring .....	4
2.2 Musikkmateriale: Fragmenter fra komposisjonsprosessen. ....	6
2.2.1 Sammenhengen mellom organisering, komposisjon og fremførelse .....	8
2.3 Samarbeidende musikkorganisering i eksisterende praksis.....	9
2.3.1 Oppbevaring .....	10
2.3.2 Overlevering .....	10
2.3.3 Type kommunikasjon.....	11
2.3.4 Bruk av IKT-programmer og dokumenter.....	11
2.4 Problemer med eksisterende praksis .....	11
<b>3. TEORETISK RAMMEVERK.....</b>	<b>13</b>
3.1 CSCW: Systemer for datastøttet samarbeid.....	13
3.2 Klassifisering av CSCW-systemer .....	15
3.3 Klassifisering av nettverksbaserte musikkssystemer .....	16
3.4 Common Information Space .....	18
3.5 Awareness .....	19
3.6 Samarbeidsmekanismer .....	20
3.7 Articulation work.....	22
<b>4. TIDLIGERE FORSKNING.....</b>	<b>24</b>
4.1 IKT-anvendelse i samarbeidende musikkproduksjon .....	24
4.2 Posisjonering av studiet .....	26
<b>5. FORSKNINGSSPØRSMÅL.....</b>	<b>30</b>
5.1 CMO: et unikt Common Information Space? .....	31
5.2 Klientapplikasjon versus webgrensesnitt .....	32
<b>6. FORSKNINGSTILNÆRMING OG METODEVALG .....</b>	<b>34</b>
6.1 Designforskning.....	34
6.1.1 Designforskning på informasjonssystemer .....	35
6.2 Kvalitative metoder .....	37
6.3 Evaluering: Groupware Walkthrough.....	40
6.3.1 Målekriterier for brukskvalitet.....	46
<b>7. DESIGN AV COLLABORATIVE MUSIC ORGANIZER .....</b>	<b>47</b>
7.1 Valg av teknologi og språk.....	47
7.1.1 Distribuert organisering via klient-/serverarkitektur.....	48
7.2 Prosessmodell .....	49
7.2.1 Analyse.....	50
7.2.2 Design .....	53
7.2.3 Koding og implementering .....	57

7.2.4	Evaluering .....	57
7.3	Designvalg: Gjennomgang av Collaborative Music Organizer .....	58
7.3.1	Innlogging .....	59
7.3.2	Oversiktsbilde .....	60
7.3.3	Prosjektutforskeren .....	61
7.3.4	Chat-vinduet .....	66
7.3.5	Kommentarvinduet.....	69
7.3.6	Loggvinduet .....	70
7.3.7	Tilleggsmomenter om designvalgene .....	71
7.4	Klassifisering av Collaborative Music Organizer .....	73
7.5	Oppsummering.....	74
<b>8.</b>	<b><u>EVALUERING.....</u></b>	<b>75</b>
8.1	Scenario #1: Diskusjon om komposisjon .....	76
8.1.1	Scenariobeskrivelse .....	76
8.1.2	Analysediagram: Diskusjon om komposisjon.....	77
8.1.3	Analyse av resultatene.....	78
8.2	Scenario #2: Endring av komposisjon .....	80
8.2.1	Scenariobeskrivelse .....	80
8.2.2	Analyse av resultatene.....	81
8.3	Scenario #3: Overlevering av musikkmateriale .....	84
8.3.1	Scenariobeskrivelse .....	84
8.3.2	Analyse av resultatene.....	85
8.4	Scenario #4: Arrangere øving .....	87
8.4.1	Scenariobeskrivelse .....	87
8.4.2	Analyse av resultatene.....	88
8.5	Resultater fra Groupware Walkthrough.....	89
<b>9.</b>	<b><u>KONKLUSJON.....</u></b>	<b>92</b>
9.1	Evaluering av studiet .....	94
9.2	Videre forskning og fremtidige utfordringer .....	96
	<b><u>LITTERATURLISTE.....</u></b>	<b>98</b>

## Figurer og tabeller

FIGUR 1 – KLASSIFISERING AV COLLABORATIVE MUSIC ORGANIZER BASERT PÅ RODDEN (1991) .....	16
FIGUR 2 – BARBOSAS 4 KATEGORIER PLASSET I RODDENS KLASSIFISERINGSSYSTEM (BARBOSA, 2006, P. 42). .....	28
FIGUR 3 – KOMPONENTENE I GROUPWARE WALKTHROUGH. ....	41
FIGUR 4 – ARBEIDSFLYTMODELL. ....	42
FIGUR 5 – ANALYSEDIAGRAM AV SCENARIOET DISKUSJON OM KOMPOSISJON .....	45
FIGUR 6 – SOFTWARE PROCESS MODEL FOR UTVIKLINGEN AV COLLABORATIVE MUSIC ORGANIZER. ....	50
FIGUR 7 – EER-MODELL AV DATABASEN. ....	54
FIGUR 8 – ABSTRAHERT UML-MODELL SOM VISER SYSTEMARKITEKTUREN.....	56
FIGUR 9 – SKJERMBILDE AV INNLOGGINGSVINDUET. ....	59
FIGUR 10 – SKJERMBILDE SOM VISER HELE PROGRAMMET ETTER INNLOGGING. ....	60
FIGUR 11 – SKJERMBILDE AV PROSJEKTUTFORSKEREN.....	61
FIGUR 12 – SKJERMBILDE AV VERSJONSKONTROLLEN.....	64
FIGUR 13 – UTSNITT AV PROSJEKTUTFORSKEREN.....	65
FIGUR 14 – SKJERMBILDE AV CHAT-VINDUET. ....	66
FIGUR 15 – SKJERMBILDE AV BRUKERINFORMASJONSVINDUET. ....	68
FIGUR 16 – SKJERMBILDE AV KOMMENTARVINDUET. ....	69
FIGUR 17 – SKJERMBILDE AV LOGGVINDUET. ....	70
FIGUR 18 – ANALYSEDIAGRAM MODELLERT AV SCENARIO #1. ....	77
TABELL 1 – SAMARBEIDSMEKANISMENE SOM TAR STED I SAMARBEID. (GUTWIN & GREENBERG, 2000). ....	21
TABELL 2 – BARBOSAS 4 KATEGORIER.....	27
TABELL 3 – SCENARIOBESKRIVELSE. (PINELLE & GUTWIN, 2002).....	43
TABELL 4 – BRUKERKRAVENE TIL COLLABORATIVE MUSIC ORGANIZER. ....	51
TABELL 5 – I HVILKEN GRAD SAMARBEIDSMEKANISMENE ER STØTTET I COLLABORATIVE MUSIC ORGANIZER. ....	90

## **Forord**

Masteroppgaven i informasjonsvitenskap har vært en krevende prosess med flere utfordringer og mye arbeid. Jeg kunne ikke ha utført dette arbeidet uten tilgang til eksterne ressurser. Først og fremst vil jeg takke min veileder, Frode Guribye, for konstruktiv kritikk og pragmatiske innspill. Jeg er også takknemlig for at IT-avdelingen ved Universitetet i Bergen bevilget meg en fast IP-adresse for serveren som er satt til for å kommunisere med Collaborative Music Organizer.

Følgende personer takkes for deres verdifulle bidrag til dette studiet: Kristian Fyllingsnes, Eyolf Nysæther, Jarle H. Olsen, Jon Martin Skauge, Jostein Fonnes, Ronny Heimdal, Peter H. Sæverud, Tomas Fordelsen, og ikke minst Øyvind Kristiansen for innspill, så vel som utenomfaglige samtaler på lesesalen.

Avslutningsvis vil jeg takke Marita, min familie og mine venner for emosjonell støtte gjennom studieprosessen.

## 1. Introduksjon

Musikk har vært en betydelig del av menneskelige kunstformer i flere tusen år. I dag kan man sjelden slå på radioen eller fjernsynet uten å erfare at musikk er en vesentlig del av programmene som sendes over disse mediene. Tar man en busstur er det vanlig å observere mennesker som hører på musikk gjennom iPod-en eller mobiltelefonen sin. Musikken disse menneskene hører på er et resultat av samarbeid. I enhver musikkproduksjon har flere aktører medvirket for å skape det kunstneriske sluttproduktet: musikere komponerer og spiller musikken, lydteknikere prosesserer lydbildet og produsenter godkjenner og kvalitetssikrer komposisjonene. De er alle ledd i et stort musikk samarbeid. I dette samarbeidet er organisering av musikk materialet, det være seg lydfiler, musikkmedium, skisser, lyrikk eller noter, en avgjørende del.

Anvendelsen av informasjons- og kommunikasjonsteknologi (IKT) i ulike arbeidskontekster har blomstret gjennom slutten av det tyvende århundre. Musikkbransjen har også erfart dette. Digitale studioer, opptaksmaskiner, avspillingsmedier, internett og nettverk har blitt integrert i musikkproduksjonen. Forskingen på å støtte samarbeidende komposisjon og fremførelse gjennom anvendelse av IKT begynte allerede på 70-tallet (Barbosa, 2003, p. 53). Hovedfokuset har lagt på IKT-systemer som støtter musikkensembler i å komponere eller fremføre musikk uten å befinne seg på samme sted. Forskningsmotivet har ofte vært å forbedre eller forenkle det eksisterende samarbeidet. I dette studiet skal det rettes fokus mot et annet betydningsfullt område i relasjon til samarbeidende musikkproduksjon. Området kan omtales som distribuert samarbeidende organisering av musikk materiale og omfatter hvordan samarbeidende komponister oppbevarer sitt musikk materiale, overleverer musikk materiale seg i mellom, og kommuniserer om musikk materialet gjennom komposisjonsprosessen – uten at de trenger å befinne seg på samme sted.

Målet for dette studiet er å tilby en IKT-løsning som kan bidra til å støtte samarbeidende komponister, musikere, lydteknikere og produsenter i organiseringen og oppbevaringen av deres kollektive musikk materiale. Implisitt i denne IKT-løsningen skal det tilrettelegges for at deltakerne kan samarbeide fra distribuerte lokasjoner. IKT-løsningen har fått navnet «Collaborative Music Organizer» (CMO).

Studiet er begrenset til feltet *Computer Supported Cooperative Work* (CSCW). Dette feltet er opptatt av hvordan datamaskiner kan benyttes for å forenkle eller forbedre gruppesamarbeid mellom flere mennesker (Bannon & Schmidt, 1989). Majoriteten av teoriene og rammeverkene som har blitt benyttet i dette studiet er etablerte konsepter innenfor CSCW-feltet. *Human-Computer Interaction* (HCI) er et annet mye anvendt felt i IKT-forskning. HCI fokuserer på design, konstruksjon, evaluering og implementering av interaktive datasystemer der mennesker er involvert (Hewett et al., 1996, p. 5). CSCW har sprunget ut fra HCI, men sistnevnte fokuserer mindre på samarbeid, og CSCW kan derfor gi en rikere innfallsvinkel når samarbeid utgjør studiets kjerne. For å øke innsikten i hvordan tradisjonelt musikk-samarbeid foregår, har det i tillegg blitt benyttet musikkstudier med en sosiologisk tilnærming.

## **1.1 Studiets struktur**

I kapittel 2 blir bakgrunnen for studiet presentert. Etter en kort begrepsforklaring blir det beskrevet hvordan komposisjon av musikk utføres i samarbeid og hvordan komposisjonsprosessen medfører at det oppstår musikkmateriale. Dette leder videre til spørsmål om hvordan musikkmateriale oppbevares og overleveres, og hvordan deltakerne på en komposisjon kommuniserer om sitt felles musikkmateriale gjennom komposisjonsprosessen. Etterpå blir disse spørsmålene besvart med en analyse som er gjort på bakgrunn av kvalitative intervjuer med komponister. Her trekkes frem funn som viser hvordan samarbeidende organisering av musikkmateriale foregår i eksisterende praksis. Hensikten med intervjuene var å finne ut hvorvidt det eksisterte problemer eller utfordringer innenfor den eksisterende praksisen. Intervjuene kan i denne sammenhengen betraktes som en forstudie innenfor studiets problemområde. Kapittel 3 omhandler det teoretiske rammeverket som har blitt benyttet i studiet. Her presenteres viktig teori fra CSCW-feltet. I kapittel 4 trekker jeg frem tidligere forskningsbidrag innenfor datastøttet samarbeidende musikkproduksjon. Her blir det også gitt et kort innblikk i tidligere forskningstendenser; noe som bidrar til å posisjonere dette studiet. I kapittel 5 beskrives forskningsspørsmålene som er relevante i dette studiet, og i kapittel 6 blir det vist hvordan jeg har gått frem for å løse disse spørsmålene. Kapittel 6 inneholder en beskrivelse av designforskning, kvalitative metoder og valg av evalueringsrammeverk. Designet av CMO og designvalgene som er foretatt under utviklingen av denne applikasjonen blir presentert i kapittel

7. Dette omfatter også en gjennomgang av prosessmodellen som er brukt, valg av teknologi og språk, brukerkrav, databasemodellering, systemarkitektur og en gjennomgang av brukergrensesnittet og hvordan det teoretiske rammeverket (fra kapittel 3) gjenspeiles her. Evalueringen av CMO blir gjennomgått i kapittel 8. Her blir det vist hvordan evalueringen er utført gjennom fire scenarioer, etterfulgt av en analyse av funnene som ble samlet inn. I det siste kapitlet presenteres konklusjonen, en evaluering av studiet og forslag til videre forskning.



## 2. Bakgrunn

Hvis man ønsker å forbedre en eksisterende praksis ved hjelp av IKT, er det essensielt at man identifiserer sider ved praksisen som kan være gjenstand for forbedring. I dette studiet har fokuset lagt på hvordan samarbeidende organisering av musikkmateriale foregår, og om det finnes problemer eller utfordringer her. Men hva er musikkmateriale? Hvordan oppstår det? I dette kapittelet vil jeg først presentere en begrepsforklaring, etterfulgt av en beskrivelse av hvordan en samarbeidende komposisjonsprosess resulterer i musikkmateriale. Etterpå stilles følgende spørsmål: Hvordan fungerer den eksisterende praksisen for organisering av felles musikkmateriale blant samarbeidende komponister? For å besvare spørsmål relatert til samarbeidende komposisjon og organisering har jeg utført flere intervjuer med aktører innenfor musikkkyrket. Denne datainnsamlingsmetoden har vært betydningsfull for utforskning av studiets problemområde, samt for å skape en empirisk innfallsvinkel til studiet. Funnene fra intervjuene har bidratt til økt forståelse for hvordan dette fenomenet forekommer i eksisterende praksis, og det er vesentlig å inkludere en analyse av funnene i bakgrunnskapittelet. En oversikt over funnene er inkludert i appendiks B. Som supplement til funnene fra dette studiets intervjuer, vil jeg også presentere funn fra andre studier. Avslutningsvis vil jeg trekke frem problemer og utfordringer ved den eksisterende praksisen.

### 2.1 Begrepsforklaring

Det er noen begreper som bør forklares tidlig da de repeteres gjentatte ganger i løpet av tekstens forløp.<sup>1</sup>

Begrepet *musikkmateriale* omfatter det materialet som er relatert til utarbeidelse av musikalske komposisjoner i en samarbeidskontekst, for eksempel: musikkopptak (som kan avspilles via et medium), noter, settlister og lyrikk. Musikkopptak og noter kan være representasjoner av komposisjonens introduksjon, vers, refreng, riff<sup>2</sup> eller melodier, og så videre. Det kan være en

---

<sup>1</sup> Studiet er skrevet på norsk. Engelske begreper blir imidlertid benyttet dersom disse gjelder etablerte konsepter, teorier, metoder eller rammeverk innenfor informasjonsvitenskap. Oversettelsene fra norsk til engelsk, eller omvendt, finner man i fotnoter.

<sup>2</sup> Et riff er en musikalsk figur som repeteres gjentatte ganger.

hel komposisjon, men kan også være fragmenter av komposisjonen som kan videreutvikles til å bli en ferdig komposisjon. En settliste er en oversikt over låtene som skal fremføres i løpet av en konsert. Begrepet kommer fra at man ofte deler inn en konsert i flere perioder eller sett.

Menneskene som inngår i studiets problemområde ble innledningsvis omtalt som musikere, komponister, lydteknikere og produsenter. Disse rollene kan imidlertid være overlappende. For å forenkle representasjonen av disse rollene i dette studiet, blir rollene tidvis referert til som *samarbeidende komponister* i stedet for å nevne alle rollene hver gang. Men implisitt i denne representasjonen kan altså komponistene også være musikere, lydteknikere eller produsenter.

Konseptet *MIDI* er mye brukt innenfor musikkmiljøet, både i sammenheng med MIDI-filer, MIDI-kontroller og MIDI-komposisjoner. Akronymet betyr «musical instrument digital interface» og er en standard som brukes for å kontrollere musikkenheter (for eksempel en synthesizer). Standarden muliggjør at lyd som skapes på en synthesizer, kan lagres som en sekvens og overføres for å spilles av på en annen synthesizer. Alle synthesizere, lydkort og samplere som støtter MIDI kan i prinsippet skiftes ut med andre MIDI-enheter. MIDI skiller seg fra wave ved at lydene befinner seg i lydbanken til synthesizeren (eller musikkenheten) og startes ved hjelp av et kall i en MIDI-fil. MIDI-filene inneholder kun referanser til lydbanken og informasjon om hver tone som skal spilles, og dette gjør at MIDI-filer er veldig plassbesparende i forhold til wave-filer (Rubinstein, 1995, pp. 287-288). Wave er en forkortelse av Waveform audio format, og er en standard for å lagre ukomprimert lyd på PC (Digital Audio Working Group, 2005, pp. 18-19).

Begrepet *distribuert organisering* er del av tittelen på studiet. Dette begrepet har imidlertid to forskjellige betydninger. Begrepet betegner både oppbygningen av en distribuert systemarkitektur, samt hvilke implikasjoner distribuert organisering har for aktørene som deltar i samarbeidet. En distribuert systemarkitektur kan karakteriseres som et system der informasjonsprosesseringen blir fordelt over flere datamaskiner. Sentraliserte systemer er en motsetning av dette, og her blir informasjonsprosesseringen utført av en sentralisert maskin (Sommerville, 2007, p. 267). Å samarbeide distribuert, på den annen side, betyr at aktørene i et gitt samarbeid har mulighet til å interagere fra fjerne områder. Dette innebærer med andre ord at

de ikke trenger å befinne seg på samme sted under samarbeidet. Distribuert organisering trenger ikke nødvendigvis å innbefatte bruk av datamaskiner, men i dette studiet vil bruk av datamaskiner være vesentlig. (Distribuerte systemer blir ytterligere belyst i seksjon 7.1.1).

*Fremførelse*<sup>3</sup> er et gjennomgående begrep i teksten. I dette studiet omfatter begrepet situasjoner der flere mennesker spiller et musikalsk verk sammen. Fremførelsen trenger dog ikke å forekomme foran et publikum, men innebærer at menneskene enten spiller tidligere komponert musikkmateriale eller improviserer sammen i sanntid. Fremførelsen kan for eksempel forekomme i studio under en opptakssituasjon eller på en øving. Grunnen til denne forståelsen av begrepet, er at mye av forskningen på CSCW og musikkssystemer har vært nettopp å simulere samarbeidende fremførelse av musikk (se kapittel 4) og fremførelsesbegrepet i denne forskningen har vært vidt nok til å omfatte både improvisasjon og samspill (både med og uten publikum).

## 2.2 Musikkmateriale: Fragmenter fra komposisjonsprosessen.

Å komponere har blitt definert som (a) «produsere, frembringe (kunstverk, særl. litterært verk ell. maleri) slik at enkelthetene danner et harmonisk hele» og (b) «(om musiker) sette musikk til (en tekst); fortone (II); tonesette; oppfinne og utarbeide (et musikalsk verk)» (Kunnskapsforlaget, 1975-2008). Komposisjon av musikk kan i dette studiet karakteriseres som en prosess der man setter sammen flere deler til et hele<sup>4</sup>. Delene kan være harmonier, melodier, sangtekster og rytmer. Disse delene kan videre fordeles utover ulike stemmer, for eksempel et piano, en bass eller et trommesett. I en musikkssammenheng er målet å skape et ferdig musikkprodukt, for eksempel en poplåt eller et symfonisk verk. En komponist kan komponere et musikalsk verk alene, men i en del sammenhenger vil det være aktuelt med et samarbeid. Alle intervjuobjektene i dette studiet hadde erfaring fra samarbeidende komposisjon, og de nevnte flere ulike fremgangsmåter for å komponere sammen. Fremgangsmåtene innbefattet blant annet: improvisasjon, bruk av forhåndslagde ideer, brainstorming og utbygging av tidligere komponerte

---

<sup>3</sup> Eng.: Performance

<sup>4</sup> For mer omfattende definisjoner, se *Defining music* (Randell, 1987) og *The Process of Musical Creation: Interviews with Eight Composers* (Bennet, 1976).

riff. Ifølge Alperson (2008) kan improvisasjon (i grupper) utføres uten at noe musikalsk arbeid har blitt gjort på forhånd, men kan også utføres der musikerne improviserer over allerede komponert musikalsk arbeid (p. 48), for eksempel et akkordskjema.

Når musikksamarbeidet har resultert i en hel del musikkmateriale (det vil si: harmonistrukturer, riff, rytmer, vers, etc.) må deltakerne gjennomgå en selektiv prosess for å velge ut hva som skal brukes i den endelige komposisjonen. Hvilke deler som skal inkluderes i det musikalske sluttproduktet blir avgjort gjennom diskusjon mellom de deltakende aktørene, eller vedtatt av bandleider eller hovedkomponist (Boespflug, 1999, p. 35).

Samarbeidende komposisjon baserer seg altså ikke på en enkeltstående metode. Fremgangsmåtene som blir brukt i et musikksamarbeid varierer både i forhold til musikkjanger og i forhold til musikernes erfaringer (Boespflug, 1999, p. 34;37). I pop-ensembler kan for eksempel bassisten og trommeslageren møtes alene (uten engasjement fra øvrige bandmedlemmer) for å utvikle et rytmisk fundament som senere kan bli utforsket i plenum (*ibid*). Dette samsvarer med samtlige av intervjuobjektene oppfatninger. De mente det ikke var nødvendig at alle medlemmene av et band måtte være tilstede under en økt med samarbeidende komposisjon. Innenfor punksjangeren brukes det fremgangsmåter som skiller seg fra popsjangeren. Her kan komposisjonene bli utviklet gjennom konserter, der både bandet og publikum er med på utformingen av det som blir den endelige komposisjonen (Lewis, 1988, p. 39). I improvisatoriske ensembler (for eksempel en jazztrio) kan improvisasjonen mellom deltakerne medføre at det oppstår ideer som senere kan bli grunnlaget for nye komposisjoner (Boespflug, 1999, p. 35)

Komposisjon av musikk er også basert på regler og musikkteori (Zeitlin & Goldberger, 2002). Når musikere komponerer må de ta hensyn til toneart, tempo, harmonilære, rytmikk, genre og ikke minst deltakernes preferanser. Dette krever at de er selektive når de improviserer eller legger til musikk på en eksisterende komposisjon. Likevel kan en idé som ikke passer i en spesifikk sammenheng fordi den strider med noen av de overnevnte momentene, vise seg å bli nyttig i en annen sammenheng på et senere tidspunkt. Derfor kan det være hensiktsmessig å oppbevare den.

Utviklingen av en komposisjon i samarbeid kan utfolde seg over større tidsperioder og flere arbeidsøkter. Musikk materialet som utvikles i løpet av en arbeidsøkt bør derfor ivaretas slik at samarbeidsdeltakerne kan videreutvikle musikk materialet i etterfølgende arbeidsøkter. Dette innebærer at musikk materialet er tilgjengelig for samarbeidsdeltakerne dersom de har behov for å oppdatere seg på det som har blitt produsert tidligere. Musikk materialet må med andre ord oppbevares.

Et komposisjonssamarbeid kan altså føre til at det blir mye musikk materiale i omløp. Det kan også føre til et behov for oppbevaring og utvelgelse av komponert musikk materialet. Dette åpner for flere interessante spørsmål: Hvor oppbevarer man musikk materialet som blir komponert, men ikke tatt i bruk? Hva skjer med uferdige komposisjoner? Hvordan overleveres relevant musikk materiale mellom deltakerne på et musikkprosjekt, og hvordan kommuniserer samarbeidende komponister om komposisjonene? Kan et informasjonssystem forbedre disse prosessene? I de etterfølgende seksjonene skal jeg gå nærmere inn på disse spørsmålene, men først skal det redegjøres kort for sammenhengen mellom organisering, komposisjon og fremførelse.

### **2.2.1 Sammenhengen mellom organisering, komposisjon og fremførelse**

Fremførelse kan, som nevnt i begrepsforklaringen, innebære samspill eller improvisasjon mellom flere musikere (både med og uten publikum). Både under den pågående komposisjonsprosessen og etter at en komposisjon er ferdigutviklet, kan det være ønskelig å fremføre musikk materialet

Et eksempel på fremførelse av musikk materiale under den pågående komposisjonsprosessen kan for eksempel være en gruppe som improviserer over et tidligere komponert parti. Improvisasjonen kan videre medføre at det oppstår nytt musikk materiale (som nevnt i seksjon 2.2). Etter at en komposisjon er ferdigutvikling kan det være ønskelig å fremføre musikk materialet i en konsertsammenheng. Før gjennomføringen av en konsert vil det være aktuelt å ha organisert musikk materiale slik at gruppen som skal holde konserten vet hva de skal fremføre. Dette innebærer at alle medlemmene av gruppen har tilgang til og informasjon om aktuelle musikkopptak, noter, sangtekster og settlister.

Organisering av musikkmateriale (i en samarbeidskontekst) kan være nødvendig for å kunne oppbevare eksisterende musikkmateriale og utveksle nytt musikkmateriale både mens komposisjonsprosessen pågår, og etter at komposisjonen er ferdig. Dersom en individuell deltaker har komponert nytt musikkmateriale som han ønsker å sy sammen med eksisterende musikkmateriale, må dette musikkmaterialet overleveres til de andre deltakerne slik at de kan gjøre seg kjent med materialet. Dette er en forutsetning for at utvelgelsen av musikkmaterialet, som skal bli del av sluttproduktet, skal kunne gjennomføres i et samarbeid.

I seksjon 4.2 vil sammenhengen mellom organisering, komposisjon og fremførelse tydeliggjøres ytterligere.

### **2.3 Samarbeidende musikkorganisering i eksisterende praksis**

Hvordan fungerer den eksisterende praksisen for samarbeidende organisering av musikkmateriale?

Å *organisere* kan løst defineres som: «ordne, innrette (et hele bestående av forskjellige deler, særl. av individer, selskaper el. lign.) for å oppnå samvirke om et felles mål» (Kunnskapsforlaget, 1975-2008). I utformingen av dette studiets intervjuer har noen områder vært av spesiell interesse. Disse områdene ble ansett som sentrale når det gjelder å finne ut hvordan samarbeidende organisering foregår i eksisterende praksis. Områdene kan inndeles i *oppbevaring* av musikkmateriale og *overlevering* av musikkmateriale mellom flere deltakere. Det ble antatt at disse områdene er viktig i et samarbeid der flere deltakere må ordne og innrette ulike deler for å oppnå samarbeid mot et felles mål. *Kommunikasjon* er det tredje området som har blitt undersøkt. Diskusjoner er en del av den samarbeidende prosessen mellom musikere (Boespflug, 1999, p. 36), og diskusjoner innebærer nødvendigvis kommunikasjon. Diskusjonene kan blant annet være relatert til arbeidsprogresjon, mål eller planlegging av neste arbeidsøkt, koordinering av oppgaver, presentasjon av nye ideer og uenigheter om uferdige komposisjoner.

En av intensjonene med å undersøke disse områdene har vært å finne ut hvorvidt områdene er problemfrie i eksisterende praksis, eller om det eksisterer problemer innenfor områdene som kan

forbedres ved bruk av et informasjonssystem. Dersom det finnes utfordringer eller forbedringspotensial innenfor samarbeidende organisering av musikkmateriale, bør spørsmål relatert til disse områdene gi antydninger om dette.

Basert på de kvalitative intervjuene og forskningslitteratur, skal det redegjøres for noen av de grunnleggende momentene som er funnet.

### **2.3.1 Oppbevaring**

Musikkmateriale (både ferdig og uferdig) blir oppbevart for senere bruk ved hjelp av følgende måter: (1) musikken spilles inn på et opptaksmedium (Boespflug, 1999, p. 36), (2) noteres på papir vha. musikknotasjon<sup>5</sup> (Bellini, Fioravanti, & Nesi, 1999, p. 26), (3) komponisten stoler på sin egen hukommelse og «lagrer» musikkmateriale midlertidig i hjernen, (4) noteres på papir ved å bruke egenutviklet syntaks, (5) programmeres i et MIDI-program.

Opptaksmedium (1) representerer i denne sammenhengen bruk av minidisk, diktafon eller lydprogram på datamaskin.

### **2.3.2 Overlevering**

Overlevering av musikkmateriale, fra en deltakende komponist til en annen, forekommer gjennom fysiske overleveringer eller digitale overleveringer. Fysiske overleveringer innebærer å levere noter eller cd, å fremføre en komposisjon ved hjelp av instrument eller avspiller, eller å vise skisser av en komposisjon. Dette krever at minst to komponister møtes ansikt til ansikt, der den ene er en sender og den andre en mottaker. Digitale overleveringer omfatter å sende komposisjonene til hverandre ved bruk av e-post eller å tilby muligheter for nedlasting av komposisjonene gjennom en webside eller server. Et intervjuobjekt meddelte at han brukte egen FTP-server i sammenheng med internasjonale musikere, der hver deltaker fikk tilsendt brukernavn og passord for å få tilgang til musikkmateriale.

---

<sup>5</sup> Uttrykkes også som «vestlig europeisk musikknotasjon», og er betegnelsen på et notesystem som brukes for å representere musikk gjennom symboler.

### 2.3.3 Type kommunikasjon

Hvordan kommuniserer deltakerne med hverandre om komposisjonene de samarbeider på? Her viste det seg å være en blandet praksis. Fremgangsmåtene innbefattet blant annet bruk av telefon, e-post, personlig møte mellom to eller flere, Instant Messaging<sup>6</sup> eller diskusjoner i øvingslokalet. På spørsmål om hvilke kommunikasjonsformer som ble foretrukket, var det interessant at dette varierte blant alle intervjuobjektene. Muntlig kommunikasjon (både ansikt til ansikt og telefon) og e-post utmerket seg imidlertid som favorittmedium blant flere av dem.

### 2.3.4 Bruk av IKT-programmer og dokumenter

Det var stor variasjon i hvilken programvare som ble benyttet til komposisjon av musikkmateriale. Sibelius, Samplitude, Wavelab, Cakewalk Sonar, Cubase, Pro Tools og Garage Band er noen eksempler. På spørsmål om hvilke typer dokumenter (tekstlig) intervjuobjektene var vant til å forholde seg til i sin musikkpraksis, svarte samtlige settlister. I tillegg ble det blant annet nevnt noter, sangtekster, kontrakter, papirutkast, tabulatur<sup>7</sup>, blekker<sup>8</sup>, og regnskapsdokumenter.

På slutten av hvert intervju ble det presentert et scenario til intervjuobjektene. Scenarioet beskrev den grunnleggende funksjonaliteten til CMO (som var gjenstand for utvikling da). På et hypotetisk spørsmål om hvor (det vil si: fra hvilket sted) aktørene trodde de ville tatt i bruk systemet for å organisere felles musikkmateriale, svarte samtlige at de ville brukt det hjemme hos seg selv. De som drev eget studio svarte at de ville bruke det både hjemmefra og fra studioet. Dette momentet omtales nærmere i seksjon 3.2 om klassifisering av CSCW-systemer.

## 2.4 Problemer med eksisterende praksis

Organiseringen av musikkmateriale i eksisterende praksis er relativt ad hoc. Den eksisterende praksisen vitner om at både oppbevaring og overlevering av musikkmateriale, og kommunikasjon om musikkmateriale skjer på mange forskjellige måter, som kan lede til flere

---

<sup>6</sup> Også omtalt som direktemeldingsfunksjonalitet

<sup>7</sup> Notasjonssystem. Hovedsakelig for strenginstrumenter. Viser fingerposisjoner i forhold til streng på instrumentet.

<sup>8</sup> Slanguttrykk for noteark eller notehefter



potensielle problemer: noe musikalsk arbeid blir glemt, noe musikalsk arbeid forsvinner, enkelte deltakere kan lett ekskluderes fra en overlevering av en komposisjon, det kan kreves mye tid å oppdatere de involverte aktørene på et musikalsk prosjekt som ikke har vært utviklet kontinuerlig og fysiske innspillingsmedier kan bli ødelagt.

Denne typen problemer ble også bekreftet av flere intervjuobjekter. De nevnte blant annet opptak og cd-er som har forsvunnet, harddiskfeil og virus som har resultert i ødeleggelse av komposisjoner og tap av fysiske noteark og sangtekster. Det ble også nevnt at når intervjuobjektene hadde basert seg på hukommelsen for å huske musikkmateriale, så hadde dette ledet til at musikkmateriale ble glemt.

Det er tydelig at det eksisterer noen problemer ved den eksisterende praksisen. Et viktig spørsmål er hvordan disse problemene potensielt sett kan løses gjennom bruk av en IKT-applikasjon, og om en IKT-applikasjon kan forebedre organiseringssamarbeidet. Funn fra intervjuene viste at samtlige av intervjuobjektene anvendte IKT under komposisjonsprosessen, og samtlige hadde erfaring fra samarbeidende komposisjon av musikkmateriale. Selv om bruken av IKT i musikkproduksjon er en ny trend, er det antatt at denne bruken finner sted i større populasjoner enn det som har vært grunnlaget for datainnsamlingen i dette studiet. Før cd-en entret markedet i 1983 brukte komponister analoge medier under komposisjonsprosessen, men dette har endret seg etter hvert som digital medier ble introdusert (Digital Audio Working Group, 2005). Funnene fra intervjuene viste at musikknoter og tekst blir fremdeles skrevet på papir i dag, men samtidig benyttes IKT-baserte tekst- og notasjonsverktøy.

Samlet viser dette at det eksisterer et grunnlag for å ta i bruk et IKT-applikasjon. Man bør likevel være bevisst på at digitale medier også har ulemper; et systemkrasj kan være nok til å ødelegge materialet som er lagret på en datamaskin, og et virus kan ha negativ innvirkning på systemstabiliteten og kan medvirke til at opptak eller dokumenter går tapt.

I kapittel 5 skal jeg utdype forskningsspørsmålene som blir benyttet i dette studiet. Men først skal jeg introdusere teorien som har blitt brukt i studiet, etterfulgt av en kort presentasjon av tidligere forskning.

### 3. Teoretisk rammeverk

Dette kapittelet vil utdype det teoretiske rammeverket som blir benyttet i studiet. Dette innebærer blant annet en beskrivelse av *Computer Supported Cooperative Work*, klassifisering av CSCW-systemer, klassifisering av nettverksbaserte musikkssystemer, relasjonene til *common information spaces*, viktigheten av *awareness* og en presentasjon av *articulation work*. I tillegg skal det redegjøres for *samarbeidsmekanismene* som eksisterer i et samarbeid; et konsept som er av stor betydning for evalueringen.

#### 3.1 CSCW: Systemer for datastøttet samarbeid

Computer Supported Cooperative Work (CSCW), eller systemer for datastøttet samarbeid, er ifølge Greif (1988, p. 5) et identifiserbart forskningsområde som er fokusert på datamaskinens rolle i gruppearbeid. Bannon & Schmidt (1989, p. 360) uttaler at CSCW kan forestilles som en bestrebelse etter å forstå naturen og karakteristikken av samarbeidende arbeid med det formål å designe hensiktsmessige datamaskinbaserte teknologier til å støtte dette.

Organisering av musikkmateriale er en aktivitet som utføres både ved hjelp av IKT og uten bruk av IKT. Dette ble illustrert i forrige kapittel. En av grunnene til å anvende CSCW kan være et ønske om å støtte den eksisterende aktiviteten ved bruk av datamaskiner. Motivene for å bruke CSCW kan være flerfoldige, men ofte ønsker man å forbedre eller forsterke samarbeidet. I følge Bannon & Schmidt (1989, pp. 359-360) er det likevel essensielt at man ikke fokuserer på kun å støtte aktiviteten ved hjelp av teknologi, men at man plasserer like mye ressurser på å forhindre at bruken av datamaskiner forstyrrer eller ødelegger aktiviteten slik den allerede fungerer. En god forståelse av arbeidskonteksten må ligge til grunn for at innføring av datastøtte til den eksisterende praksisen skal gi positive resultater.

*Groupware*<sup>9</sup> er et annet begrep som viser til datasystemer som støtter gruppearbeid. Grensene mellom CSCW og groupware har vært flytende, og det har blitt hevdet at termene blir brukt som synonymer (Ellis, Gibbs, & Rein, 1991, p. 39). Et forslag til definisjon av groupware ble tidligere presentert av Ellis et al.:

---

<sup>9</sup> No.: Gruppevare.

«computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment» (1991, p. 40).

Bannon & Schmidt (1989) har imidlertid kritisert dette bildet. De mener at å avgrense groupware-begrepet til grupper av mennesker som jobber mot et felles mål, gjør at denne definisjonen blir mye mindre omfattende enn CSCW (spesielt dersom det er snakk om groupware versus CSCW som forskningsfelt). En gruppe mennesker kan sies å være tydelig definert og relativt lukket, mens samarbeid ikke bør være begrenset på denne måten. Samarbeid er, på den annen side, rikt og komplekst med mange aktører som ikke nødvendigvis kan karakteriseres som en fast gruppe, og som heller ikke nødvendigvis har et klart definert felles mål (pp. 361-363). I tillegg kritiserer de groupware for å være teknologidrevet. Dette innebærer at fokuset ligger på å løse tekniske utfordringer, noe som medfører at utviklerne ikke trenger å ta spesielt hensyn til konteksten der gruppevaren skal brukes (*ibid*). I dag er imidlertid ikke forholdet mellom begrepene særlig turbulent; begge begrepene er fremdeles populær, og selv om mange kommersielle rapporter fremdeles referer til groupware som et forskningsfelt, er CSCW blitt det foretrukne begrepet i forskningsmiljøet (Schmidt & Bannon, 1992, pp. 9-10). Sammenhengen kan uttrykkes enklere ved å beskrive CSCW som forskningsfeltet, og groupware som de tekniske systemene som er grunnlagt på teori fra forskningsfeltet (Koch & Gross, 2006, p. 2).

CMO kan karakteriseres som groupware. Målet med applikasjonen er å støtte en gruppe mennesker som jobber med et felles prosjekt (eller mål) ved å tilby et grensesnitt til et shared workspace. Det er imidlertid ikke slik at samtlige av brukerne innenfor dette studiets problemområde nødvendigvis har samme oppgaver og samme mål. Dette fordi deltakerne fra arbeidskonteksten er både musikere, lydteknikere og produsenter. Musikerne komponerer musikk, lydteknikerne prosesserer og mikser musikken, og produsentene lytter til musikken og vurderer den mot ønsket resultat, kommersielle interesser og så videre. Ulikhetene innad i de forskjellige rollene kan bedre forstås dersom man skiller mellom rollenes underordnede og overordnede mål. Et underordnet mål kan for eksempel være lydteknikerens miksing og balansering av en gitt komposisjon, eller musikerens komponering av en gitarsolo. Men selv om de underordnede målene skiller seg fra hverandre, kan man hevde at begge jobber mot et felles overordnet mål: å ferdigstille en komposisjon.

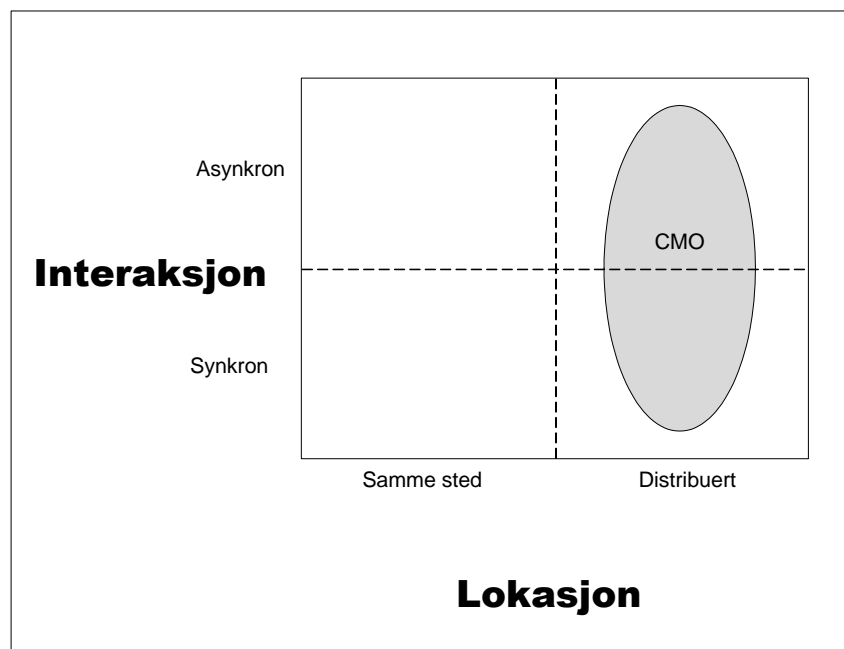
I dette studiet har både CSCW-feltet og groupware-begrepet vært av betydning. CSCW-teorien presiserer viktigheten av å forstå arbeidskonteksten man ønsker å støtte, og groupware-begrepet bidrar til å skildre applikasjonen som har blitt utviklet. Begrepet groupware blir omtalt ytterligere senere; utgangspunktet for evalueringen har vært en metode som kalles *Groupware Walkthrough* (Pinelle & Gutwin, 2002). Dette vil bli belyst i seksjon 6.3.

### 3.2 Klassifisering av CSCW-systemer

Rodden har konstruert en grafisk klassifisering av sammenhengen mellom interaksjon og lokasjon i CSCW-systemer (1991)<sup>10</sup>. Konseptet er delt inn i interaksjon (asynkron eller synkron) og på hvilken lokasjon interaksjonen finner sted (samme eller distribuert lokasjon). Konseptet er også tilrettelagt for å håndtere potensielle skjæringspunkter mellom forskjellige lokasjoner og forskjellige typer interaksjon som kan tilbydes via et CSCW-system. Og interaksjon er viktig i CSCW-systemer. Kommunikasjon er avgjørende i systemer som støtter samarbeid. Som vi så i seksjon 2.3.3, er diskusjoner om musikk materialet også en betydelig del av det eksisterende samarbeidet mellom komponister. Innenfor CSCW blir det som nevnt benyttet to begreper for å kjennetegne forskjellige typer interaksjon: synkron og asynkron. Synkron modus beskriver hvorvidt deltakernes interaksjon og samarbeid utføres i sanntid, som betyr at flere aktører er aktive i systemet samtidig. Denne modusen skiller seg fra asynkron modus ved at sistnevnte går ut på at samarbeidet mellom aktørene kan ta sted uten at alle er aktiv i systemet på samme tidspunkt (Rodden, 1991, p. 2).

---

<sup>10</sup> I følge Ellis et al. (1991) ble denne taksonomien først presentert av Johansen i boken: *GroupWare: Computer Support for Business Teams* (1988).



**Figur 1 – Klassifisering av Collaborative Music Organizer basert på Rodden (1991)**

I figur 1 er det illustrert hvordan dette forholdet reflekteres i dette studiet. Her ser man at CMO vil innbefatte både asynkron og synkron kommunikasjon, men kun fra distribuerte lokasjoner. Det er ikke dermed sagt at man ikke kan plassere to klientdatamaskiner i samme

rom og bruke CMO, men dette er ikke intensjonen.

Grunnen til dette, er at det antas at noe slikt ikke er interessant for brukerne. Gitt at applikasjonen ville fasilitert synkron samarbeidende komponering av musikk, ville dette vært en mer interessant problemstilling. Da kunne eksempelvis bassisten og gitaristen sittet hver for seg i samme rom og gjort opptak av musikkmateriale. Men med hovedvekt på organisering av musikkmateriale, ønskes det ikke å ta høyde for at synkron eller asynkron interaksjon (eller samhandling) på samme sted vil være attraktivt. Dette samsvarer også med funnene fra intervjuene (se seksjon 2.3.4).

### 3.3 Klassifisering av nettverksbaserte musikkssystemer

Weinberg (2002, pp. 352-354, 2005, pp. 26-28) har presentert en klassifisering av nettverksbaserte musikkssystemer. Denne klassifiseringen er inndelt i fire kategorier. I hver kategori måles det hvilken grad av sammenkobling og gjensidig avhengighet det finnes mellom deltakerne, samt hvilken rolle nettverkssystemet spiller i å forbedre sammenkoblingen og de sosiale relasjonene mellom deltakerne. En kort presentasjon av disse kategoriene følger:

(1) *The Server Approach* bruker nettverket for å sende og motta data fra separate brukere, men benyttes ikke til direkte tilkobling eller kommunikasjon mellom deltakerne. Denne metoden

tilbyr heller ikke muligheter for lytting eller interaksjon mellom deltakerne, og er begrenset til hver deltakers kommunikasjon med den sentrale serveren. Dette er altså en metode som ikke yter noen som helst form for synkrone samarbeidsmuligheter. Nettverket blir i hovedsak brukt til distribusjon av teknologi eller data.

(2) *The Bridge Approach* skaper en bro mellom to eller flere avstandsliggende deltakere slik at de kan musisere og improvisere som om de var på samme fysiske sted. Dette er en form for imitasjon av det eksisterende gruppesamarbeidet som finner sted blant musikere og komponister. Poenget er ikke å berike samarbeidet med ekstra muligheter, men heller tilby en teknologisk løsning slik at gruppesamarbeidet kan ta sted på samme måte som før uten at deltakerne befinner seg i samme rom.

(3) *The Shaper Approach* tar sikte på å disponere nettverkets sentrale system til å generere musikkmateriale. Systemet tilbyr videre muligheter for å la deltakerne utforme og modifisere det genererte materialet. Konseptet går kort fortalt ut på å (a) endre på parametre for så at systemet på bakgrunn av dette kan generere musikk, eller (b) bruke allerede generert musikk og endre på parametre for å endre uttrykket til den genererte musikken. Systemet etablerer imidlertid ingen gjensidig avhengighet mellom brukerne på den måten som *The Bridge Approach* gjør. Konseptet skaper likevel en større grad av sosiale relasjoner mellom deltakerne, siden hver av dem kan lytte til og svare på andres komposisjoner.

(4) *The Construction Kit Approach* tilbyr høyere nivåer av sammenkobling mellom deltakerne ved å la dem bidra med musikk til flerbrukerkomposisjoner. Her kan deltakerne manipulere og forme deres eget og andre deltakers musikkmateriale. Dette skiller seg fra de andre klassene ved at her er aktivitetene lagt opp til komposisjonsscenarioer der flere deltakere er involvert. Man kan med andre ord delta aktivt på en komposisjon med andre brukere. Interaksjonen i slike nettverk er ofte asynkron, ettersom deltakerne laster opp sitt eget materiale online og endrer på andres materiale offline. All virksomhet skjer altså ikke online.

Disse fire kategoriene er beskrivende for de fleste nettverksbaserte musikkssystemer, selv om man må være klar over at hver kategori kan overlappe med andre kategorier. I seksjon 7.4 skal det vises hvordan CMO kan klassifiseres i forhold til Weinbergs (2002) kategorier.

### **3.4 Common Information Space**

Common Information Space (CIS) blir ofte benyttet for å omtale fenomenet der flere samarbeidende aktører anvender et felles distribuert system for å utføre både individuelle og samarbeidende aktiviteter. Dette er et av mange begreper om det samme fenomenet, og denne typen systemer har også blitt omtalt som shared workspaces, shared information spaces og common communication spaces (Bannon, 2000). Schmidt og Bannon (1992) beskriver CIS på følgende måte:

«Here the focus is on how people in a distributed setting can work cooperatively in a common information space - i.e. by maintaining a central archive of organizational information with some level of 'shared' agreement as to the meaning of this information (locally constructed), despite the marked differences concerning the origins and context of these information items» (p. 16).

Som man ser, er det viktig at det tilrettelegges for at deltakerne har mulighet til å oppnå en felles enighet om meningen til informasjonen i CIS-et, blant annet gjennom å skildre opphavet eller produsenten av informasjonen, samt konteksten der informasjonselementene kom fra. Det er ikke tilstrekkelig å tilby teknologi som muliggjør deling av informasjon; et fungerende CIS krever at deltakerne opptrer som aktivt deltakende aktører. Alle som tar del i samarbeidet må aktivt forme, tolke og diskutere meningen av informasjonsobjektene (Schmidt & Bannon, 1992, p. 20). En av de største utfordringene med CIS i forbindelse med bruk av datamaskiner, har vært nettopp å skape denne felles forståelsen for informasjonen som er tilgjengelig i aktørenes felles arbeidsområde. Spesielt dersom informasjonen er felles mellom flere mennesker med ulik bakgrunn, sjargong og kompetanse (Bannon & Bødker, 1997, p. 4; Bannon & Robinson, 1991). Noe av denne problematikken kan synliggjøre seg i dette studiet også. Musikere har gjerne en annen sjargong enn en produsent, og forståelsen av de felles informasjonsobjektene vil ikke nødvendigvis være koherent mellom dem. Det samme gjelder lydteknikere, som har mer teknisk bakgrunn og kompetanse enn musikerne. Likevel har alle tradisjonelt sett vært en del av musikkbransjen og det antas at dette ikke vil by på betydelige utfordringer. I seksjon 7.3.4 om designvalg skal jeg gå nærmere inn på dette aspektet.

CMO kan karakteriseres som et CIS i dette studiet. Forholdet mellom CMO og andre lignende systemer blir omtalt i seksjon 5.1.

### 3.5 Awareness

Awareness er et viktig konsept innenfor CSCW, spesielt i sammenhenger der applikasjoner benyttes av flere mennesker, for eksempel shared workspaces. Konseptet går ut på å vite hva menneskene rundt deg driver på med, hvor de er og hva de har gjort i det delte arbeidsområdet. Selv om det kan være lett å innhente denne informasjonen i den virkelige verden, har det vist seg at det ofte er svært vanskelig å orientere seg om dette i sanntids distribuerte systemer (Gutwin & Greenberg, 2002, p. 411). Awareness har blitt definert av Dourish & Bellotti som:

«...an understanding of the activities of others, which provides a context for your own activity. This context is used to ensure that individual contributions are relevant to the group's activity as a whole, and to evaluate individual actions with respect to group goals and progress» (1992, p. 1).

Dette innebærer at alle aktørene i et gitt samarbeid har mulighet til å orientere seg om andre aktørers handlinger som finner sted i det samme samarbeidet. Med kontekst mener Dourish & Bellotti både objektet som blir produsert i samarbeid og fremgangsmåten som blir brukt for å produsere objektet. Kontekst refererer med andre ord ikke kun til innholdet i individuelle bidrag, men også betydningen eller egenarten til de individuelle bidragene, samt hvordan bidraget er relatert til gruppens mål og progresjon (*ibid*).

Det finnes flere undergrupperinger av awareness, og *workspace awareness* (WA) kan være mest interessant i dette studiet. CMO ble tidligere karakterisert som et common information space (CIS), og et CIS kan også tolkes som å være eller bestå av et shared workspace<sup>11</sup>. WA kan defineres som: «...the up-to-the-moment understanding of another person's interaction with the shared workspace» (Gutwin & Greenberg, 2002, p. 438). Dette går ut på å kunne holde seg umiddelbart orientert om hvem som er aktiv, hvem som er tilgjengelig, hvor andre deltakere arbeider, hvilke handlinger de gjør og intensjonene med handlingene, og i tillegg omfatter dette å

---

<sup>11</sup> Se Bannon (2000) for en diskusjon om forholdet og diskrepansen mellom alle begrepene som har blitt brukt som synonymer eller tilsvarende for CIS.



være orientert om hva som har hendt eller blitt gjort tidligere (Gutwin & Greenberg, 2002, p. 439). Workspace awareness er en ekspansjon av awareness der hovedpoenget er at informasjon om interaksjonen i et shared workspace må presenteres øyeblikkelig til deltakerne. Dette effektiviserer og forenkler interaksjonen til deltakerne i workspace-et ved at de får umiddelbart informasjon om interaksjonen til sine meddeltakere og kan bruke denne informasjonen til å hjelpe hverandre, diskutere og koordinere sine oppgaver (Gutwin & Greenberg, 2002, p. 440). Hvis for eksempel en bruker A kan observere at bruker B jobber med et prosjekt C, så slipper brukeren å spesifisere prosjekt C dersom han ønsker å diskutere prosjektet med bruker B. På denne måten reduserer man anstrengelsen som må til for å interagere i workspace-et.

I dette studiet kan WA inkorporeres i CMO ved å gjenspeile hvem som er tilgjengelig i applikasjonen til enhver tid, tilrettelegge for at deltakerne kan diskutere intensjonene med handlingene sine (gjennom synkron og asynkron interaksjon), og vise hvilke handlinger som har blitt utført på det felles musikk materialet. I tillegg kan det beskrives hvem som har gjort hva, samt på hvilket tidspunkt en eventuell handling ble utført.

### **3.6 Samarbeidsmekanismer**

Gutwin & Greenberg (2000) har utviklet et konseptuelt rammeverk for å vise samarbeidsmekanismene <sup>12</sup> som finner sted i et samarbeid. Her har de identifisert syv hovedaktiviteter. Disse mekanismene må i følge Gutwin og Greenberg være tilstede for at en deltaker skal kunne utføre en oppgave i et shared workspace. Den underliggende ideen bak dette rammeverket er at mange av problemene i CSCW- og gruppevaresystemer ikke er relatert til den sosiale konteksten som systemet anvendes i, men snarere relatert til dårlig støtte for de grunnleggende basisaktivitetene som utføres av deltakerne. Hensyn til kontekst er fremdeles viktig, men når det gjelder brukskvaliteten i gruppevare, er ikke konteksten avgjørende (*ibid*).

Samarbeidsmekanismene (Gutwin & Greenberg, 2000, pp. 99-100) blir illustrert i tabell 1.

---

<sup>12</sup> Eng.: The mechanics of collaboration

Tabell 1 – Samarbeidsmekanismene som tar sted i samarbeid. (Gutwin &amp; Greenberg, 2000).

<b>Samarbeidsmekanisme:</b>	<b>Beskrivelse:</b>
<b>Eksplisitt kommunikasjon:</b>	Verbal eller skriftlig kommunikasjon er en viktig del av alle typer samarbeid.
<b>Indirekte kommunikasjon:</b>	Informasjon som tilbys om en bruker uten at han gir den fra seg selv. For eksempel informasjon som blir gitt av informasjonsobjektene han jobber med.
<b>Koordinasjon av handling:</b>	Mennesker forsøker å organisere handlingene sine slik at de ikke kommer i konflikt med andre. Noen oppgaver krever at handlinger skjer på en særskilt orden. Dårlige muligheter for koordinasjon preger systemet dersom mennesker dupliserer sine handlinger.
<b>Planlegging:</b>	Deltakerne i et shared workspace planlegger oppgaver og fordeler dem mellom seg. Systemet må tilby muligheter for å gjøre dette.
<b>Overvåkning:</b>	Flere av samarbeidsmekanismene baserer seg på at man kan overvåke handlinger og hente informasjon om sine meddeltakere. Mye av dette er awareness-informasjon.
<b>Hjelp:</b>	Samarbeidsdeltakere tilbyr hjelp til hverandre når det trengs. Grunnlaget for å få noen til å hjelpe seg kan bygge på å uttrykke behovet om hjelp gjennom eksplisitt kommunikasjon fra eller awareness om en brukers handlinger. Her er det imidlertid viktig at deltakerne forstår hva andre gjør og hvor de er i arbeidet sitt.
<b>Beskyttelse:</b>	Deltakere kan slette eller ødelegge andre sitt arbeid. Systemet og deltakerne må gå aktivt inn for at slike tilfeller ikke inntreffer.

I dette studiet fungerer samarbeidsmekanismene som veiledende krav til basisfunksjonalitet i CMO. Det er flere grunner til at samarbeidsmekanismene utgjør en fundamental del av dette studiet. For det første kan brukskvaliteten i applikasjonen forbedres betydelig, siden alle mekanismene utgjør viktige trekk for å opparbeide en felles forståelse av informasjonsobjektene i et CSCW-system. Og for det andre er samarbeidsmekanismene vesentlig i evalueringen av applikasjonen. Groupware Walkthrough (som utgjør evalueringsmetoden i studiet) baserer seg på å benytte samarbeidsmekanismene for å analysere de samarbeidende aktivitetene som foregår i systemet. Hvordan disse konseptene har blitt anvendt under designet av applikasjonen vil bli presentert i kapittel 7 om designet av CMO.

### 3.7 Articulation work

Et samarbeid består naturligvis av to eller flere aktører. For at et samarbeid skal fungere må det defineres hvem som skal gjøre hva, hvordan de skal gjøre det, når de skal gjøre det, og hvor de skal gjøre det. Et samarbeid innebærer med andre ord arbeidsfordeling, både i forhold til aktørene som utfører arbeidet og selve handlingene som blir utført (Fjuk, Smørdal, & Nurminen, 1997, pp. 1-3). Articulation work er et begrep som omhandler koordinasjonen av oppgaver mellom aktørene som inngår i et samarbeid. Ifølge Gerson & Star (1986) er articulation: «all the tasks involved in assembling, scheduling, monitoring, and coordinating all of the steps necessary to complete a production task» (p. 266). Schmidt og Bannon (1992) skildrer articulation work som en integrert del av ethvert samarbeid, og mener det kan karakteriseres som det settet av aktiviteter som er nødvendig for å forvalte «the distributed nature of cooperative work» (p. 18).

Gerson & Star (1986) uttrykker at det er umulig å forutse alle eventualiteter og tenkbare muligheter som kan inntreffe i en situasjon der samarbeidende deltakere utfører et sett av oppgaver. Man kan med andre ord ikke konstruere en formell beskrivelse av et system og garantere at det ikke vil forekomme uforutsette hendelser som er inkonsistent med den formelle beskrivelsen. Når man ikke kan forutse alle eventualiteter i et system må man tilrettelegge for at deltakerne i et samarbeid kan artikulere aktivitetene, det vil si planlegge, koordinere og overvåke, for å kunne fullføre den oppgaven de skal løse (pp. 266-267).

I dette studiet er articulation work et viktig element i organisering av musikkmateriale. Å tilrettelegge for articulation work i organiseringsaktivitetene er nødvendig for at de andre delene av musikk samarbeidet, det vil si komposisjon eller fremførelse, skal kunne gjennomføres i et CSCW-system.

Schmidt & Bannon (1992, p. 22) uttaler at konstruksjonen av et common information space (CIS) er en god fremgangsmåte for å støtte articulation work i et CSCW-system. Dette er også gjenspeilet i dette studiet; tidligere er CMO karakterisert som et CIS. I kapittel 7 om designvalg skal det vises hvordan articulation work er støttet gjennom tilretteleggelse for kommunikasjon, planlegging, overvåkning og koordinasjon i CMO. I seksjon 4.2 skal det forklares hvorfor articulation work er vesentlig for å posisjonere dette studiet, og det skal ytterligere illustreres

hvorfor organisering av musikkmateriale kan karakteriseres som articulation work. Men først skal jeg presentere tidligere forskningsbidrag innenfor CSCW og musikk.

## **4. Tidligere forskning**

Det har vært mange bidrag til CSCW og musikkproduksjon, og ofte har dette kommet i form av applikasjoner som tilbyr en innovativ innfallsvinkel gjennom ny teknologi. I introduksjonen ble det gitt en indikasjon på hvordan dette studiet kunne plasseres i forhold til tidligere forskning. CSCW-forskningen innenfor musikkproduksjon har, som man snart skal se, fokusert mye på komposisjon og fremførelse, mens organisering ikke har vært mye i søkelyset. Det bør også nevnes at forskningsbidragene innenfor musikkproduksjon ikke nødvendigvis har blitt identifisert med CSCW-feltet. Noen bruker groupware som basis, og andre har vært mer opptatt av målet med studiet fremfor å systematisere det innenfor etablerte forskningsfelt. Ifølge Barbosa (2006, p. 40) medfører dette at det er vanskelig å få oversikt over alt som har blitt gjort innenfor nettverksbasert musikkkomposisjon og fremførelse, og mange av eksperimentene forblir derfor upublisert og ukjent for forskningsfeltet.

I dette kapitlet skal det først presenteres et utvalg av forskningsbidrag til samarbeidende musikkproduksjon, og avslutningsvis vil jeg presentere en kort diskusjon om forskningstendensene innenfor forskningen på CSCW og musikkproduksjon. Dette bidrar til å illustrere hvordan dette studiet kan posisjoneres.

### **4.1 IKT-anvendelse i samarbeidende musikkproduksjon**

Et av de tidligste forsøkene på bruk av IKT til samarbeidende musikkkomposisjon og fremførelse kom på midten av 70-tallet. Dette ensemblet ble kjent som «The league of automatic music composers». Disse musikerne koblet datamaskinene sine (Commodore KIM-1-maskiner) sammen for deretter å sende komposisjoner over nettverket. Dette kan karakteriseres som et av de første forsøkene å å lage programmerbare og detaljerte musikksammenkoblinger over et nettverk på samme fysiske lokasjon (Bischoff, Gold & Horton, 1978, hentet fra Barbosa, 2006, pp. 34-36). Eksperimentet muliggjorde ikke bruk av eksisterende instrumenter, og var tilsynelatende ikke interessert i å fange en eksisterende arbeidskontekst. Dette kan karakteriseres som teknologidrevet groupware, der det primære ønsket er å bruke tilgjengelig teknologi for å skape muligheter for en ny og innovativ musikkpraksis. Eksperimentet er likevel interessant; det ble utført lenge før verken CSCW eller world wide web hadde blitt introdusert. Weinberg (2002,

p. 2) presenterer en estetisk innfallsvinkel på anvendelsen av teknologi for å skape nye muligheter for musikkkomposisjon og fremførelse (fremfor å forbedre eksisterende praksis ved bruk av teknologi). Han nevner at bruken av nettverk og teknologi til musikkfremførelse potensielt sett kan bidra til opprettelse av nye kunstneriske uttrykksformer.

I 1987 ble det første forsøket av nettverksbasert musikkfremførelse over geografisk distribuerte lokasjoner gjennomført. Seansen ble iverksatt av to komponister, Nick Collins og Phill Niblock, sammen med gruppen "The Hub" (som for øvrig bestod av medlemmer fra The league of automatic music composers). I dette eksperimentet ble to musikk-trioer linket sammen i et nettverk der de kunne kommunisere med hverandre gjennom et modem (Barbosa, 2006, p. 37).

På begynnelsen av 1990-tallet ble det aktuelt å bruke internettkommunikasjon til å berike samarbeidende komposisjon. Et resultat av disse nye mulighetene ble kalt NetJam (Latta, 1991). Dette systemet fungerte slik at deltakerne kunne samarbeide asynkront ved å sende MIDI-komposisjoner til hverandre ved hjelp av e-post. NetJam hadde en e-postliste som potensielle medlemmer kunne signere seg på, og på denne måten ble de del av informasjonsstrømmen. Moderatoren til NetJam hadde som oppgave å motta innkommende MIDI-komposisjoner, distribuere dem videre til medlemmene av e-postlisten, og til slutt arkivere komposisjonene på en FTP-server slik at nye brukere kunne få tilgang til den senere (Latta, 1991, p. 104).

I 1997 kom et betydningsfullt bidrag til samarbeidende musikkfremførelse. Dette prosjektet ble kalt TransMIDI. Systemet tilrettela for at flere musikere, som befant seg på geografisk distribuerte områder, kunne spille sammen synkront i en opprettet sesjon<sup>13</sup> gjennom MIDI-kontrolleren sin via en General Midi-protokoll (Gang, Chockler, Anker, Kremer, & Winkler, 1997).

I 1997 ble det også utviklet et system som fikk navnet FMOL: Faust Music On-Line (Jordà, 2000). Hovedtanken var å bistå samarbeidende musikkkomposisjon over nettet. Systemet var stort og hadde flere interessante aspekter. Hver aktør kunne for eksempel bruke programmet som et elektronisk musikkinstrument der de brukte musen til å dra over de virtuelle strengene på

---

<sup>13</sup> Eng.: Session

instrumentet (på samme måte man drar fingrene over en gitar eller harpe). Programmet kunne også benyttes til å modifisere andre menneskers sitt arbeid i halv dupleks kommunikasjonsmodus, som betyr at man først leser og lytter, og så svarer med å modifisere (Barbosa, 2006, pp. 52-54; Jordà, 2000; Weinberg, 2002).

Noen år senere utviklet Pazel et al. (2000) en applikasjon der brukerne kunne stille på forskjellige parametere som tonal rekkevidde, volum og valg av instrumenter, samt individuelle parametere. Etter at brukerne bekreftet valgene sine, ville applikasjonen generere musikk på bakgrunn av dette. Det behøvdtes ingen kunnskapsbakgrunn fra bruk av musikknotasjon for å anvende dette systemet.

## **4.2 Posisjonering av studiet**

For å skape et mer komplett bilde av hvor dette studiet skal plasseres, bør man se på tidligere forskning og stille seg følgende spørsmål: Hva har vært forskningstendensen innenfor CSCW i sammenheng med musikkssystemer?

Barbosa (2006) har presentert et sett av kategorier for å gruppere forskningen og eksperimentene som har blitt utført innenfor CSCW og musikkssystemer (Barbosa, 2006, pp. 42-77). Kategoriene har fellestrekk med kategoriene som finnes i Weinbergs (2002) klassifiseringssystem, men det som skiller dem er at Barbosa grupperer forskningsbidragene og eksperimentene i en CSCW-kontekst ved å rette fokus mot musikkssystemenes interaksjon og lokasjon. Dette innebærer å vurdere om musikkssystemene støtter synkron eller asynkron interaksjon, og om de støtter samarbeid på samme sted eller distribuert (Barbosa, 2006, pp. 41-42). I seksjon 3.2 ble det vist hvordan CMO kan klassifiseres i forhold til interaksjon og lokasjon ved bruk av Roddens (1991) klassifiseringssystem.

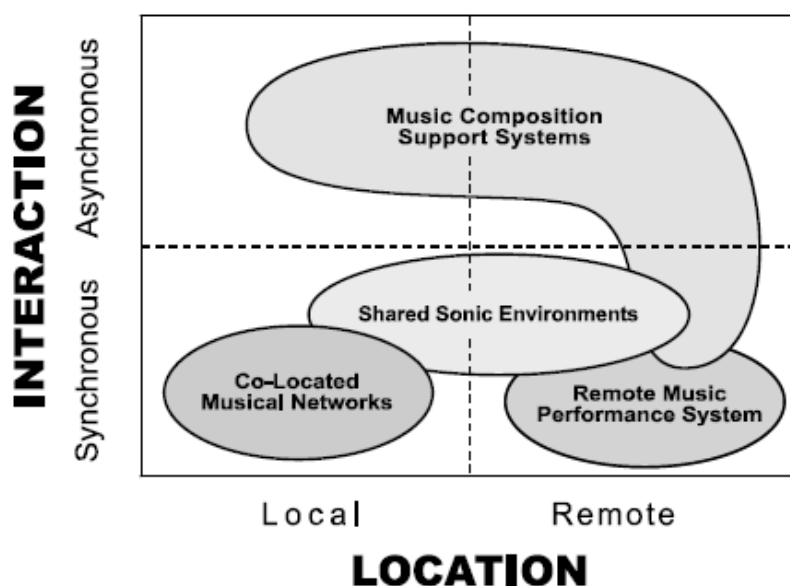
Tabell 2 – Barbosas 4 kategorier

Kategori	Beskrivelse
<b>Co-located musical networks</b>	Blir brukt i sammenhenger der grupper av musikere som interagerer i sanntid på samme fysiske sted på sine musikkinstrumenter (eller virtuelle instrumenter).
<b>Music composition support systems</b>	Blir brukt til å støtte tradisjonell musikkkomposisjon og produksjon. Støtter skriving av musikk (for eksempel noter) og opptak av musikk. Muliggjør at deltakerne kan interagere fra distribuerte steder asynkront.
<b>Remote music performance</b>	Anvendt i sammenhenger der grupper av distribuerte musikere kan improvisere og spille sammen synkront på sine musikkinstrumenter (eller virtuelle instrumenter).
<b>Shared sonic environments</b>	Klasser av applikasjoner som anvendes gjennom internett. Kategorien omfatter systemer som er åpne for allmennheten, og som kan tilby online improvisasjonsmuligheter for de som er koblet til systemet.

Kategoriene er vist i tabell 2. Ingen av kategoriene er imidlertid ment å være lukket, og en gitt applikasjon kan derfor posisjoneres under flere kategorier dersom dette er hensiktsmessig. Co-located musical networks og remote music performance innebærer at det forekommer synkron interaksjon eller spilling på deltakernes musikkinstrumenter i systemene som hører inn under denne kategorien. Dette hører sammen med begrepet fremførelse (som definert i begrepsforklaringen). Det samme gjelder shared sonic environments der hensikten er å tilby online improvisasjonsmuligheter for deltakerne gjennom internett. Music composition support systems retter fokus på komposisjon og produksjon av musikk gjennom å muliggjøre skriving og opptak av musikk. Kategoribeskrivelsene viser at komposisjon eller fremførelse er nevnte aktiviteter i hver kategori<sup>14</sup>. Videre illustrerer Barbosa (*ibid*) hvordan disse kategoriene kan plasseres i forhold til Roddens (1991) grafiske klassifisering av CSCW-systemer. I figur 2 vises det hvordan kategoriene er plassert og hvordan noen av dem overlapper med hverandre i henhold til interaksjon og lokasjon.

<sup>14</sup> CMO blir ikke plassert i noen av disse kategoriene da fokuset i dette studiet ligger på organisering av musikkmateriale





Figur 2 – Barbosas 4 kategorier plassert i Roddens klassifiseringssystem (Barbosa, 2006, p. 42).

Figur ©2006 Barbosa.

Formålet med å presentere Barbosas (2006) kategorier er å identifisere tendenser innenfor forskningen på musikkssystemer som involverer samarbeid. Av kategoribeskrivelsene i tabell 2 kan man se at organisering av musikkmateriale ikke er spesielt vektlagt. Ifølge Alperson (2008, pp. 37-38) er kjernen av den musikalske praksis selve musikeren som spiller på instrumentet sitt, og instrumentet er en vesentlig del av hvordan mennesker oppfatter musikalsk praksis. Hvor kommer så organiseringsaktiviteten inn i bildet? Organiseringen av musikkmateriale kan tolkes som å være perifere handlinger for å kunne samarbeide om å skape musikk. Det er et absolutt nødvendig og tilstedeværende aspekt av samarbeidet, men man tenker gjerne ikke på organisering som kjernen i samarbeidende musikkproduksjon. Hvordan kan man støtte de perifere handlingene som inngår i samarbeidet om å skape musikk?

Articulation work er et viktig element i organiseringen av musikkmateriale for å kunne samarbeide om å skape musikk. Samarbeidende organisering av musikkmateriale innebærer alle oppgavene som involverer tilrettelegging, koordinasjon, planlegging og overvåking av de nødvendige stegene for å utføre en samarbeidsoppgave. Stegene kan i denne sammenhengen innebære overlevering av musikkmateriale eller diskusjoner om musikkmateriale, og

samarbeidsoppgaven kan i denne sammenhengen være å skape musikk. Organisering av musikkmateriale er en vesentlig del av musikkproduksjon når flere aktører er engasjert i et samarbeid, og i dette studiet er det viktig å støtte aktivitetene som er relatert til organisering av musikkmateriale mellom samarbeidende komponister. Aktivitetene kan støttes ved å tilby muligheter for at samarbeidsdeltakerne kan utføre articulation work. Tidligere ble det nevnt at det finnes atskillige fremgangsmåter for å komponere musikk i samarbeid. Dette gjør det vanskelig å forutse alle eventualiteter som kan inntreffe under samarbeidet mellom flere komponister, og derfor blir det spesielt viktig å tilrettelegge for at eventuelle brukere av CMO kan artikulere samarbeidsaktivitetene sine. I kapittel 7 blir det vist hvordan designet av CMO reflekterer dette. I neste kapittel skal jeg presentere forskningsspørsmålene som er relevant for studiet.

## 5. Forskningsspørsmål

Poenget med forskningsspørsmål er å danne et mål for studiet: Hva er det man ønsker å finne ut? Et forskningsbidrag er sjeldent isolert fra tidligere forskning. I kapittel 3 og 4 i dette studiet er det redegjort for relevant teori og tidligere forskning. Denne redegjørelsen har bidratt til en rikere forståelse for temaet, og dette har videre ledet til en innsnevring av forskningsfokuset mot området som er gjenstand for utforskning i dette studiet. Området har blitt identifisert som *distribuert samarbeidende organisering av musikkmateriale*. Betegnelsen på området har kun blitt benyttet for å posisjonere dette studiet i forhold til tidligere studier, samt tydeliggjøre kjerneområdet i studiet.

Forskningsspørsmålene i dette studiet lyder som følger:

1. Hvordan foregår samarbeidende musikkorganisering i eksisterende praksis?
2. Hvordan kan CSCW støtte eksisterende samarbeidende musikkproduksjon og tilrettelegge for distribuert organisering av musikkmateriale?

For å besvare dette har det blitt utført følgende forskningsaktiviteter:

- Utvikling av en applikasjon som kan forbedre og forenkle den eksisterende måten for samarbeidende organisering av musikkmateriale.
- Evaluering for å finne ut hvorvidt applikasjonen fungerer som verktøy til å bistå den samarbeidende organiseringen.

Gjennom det første spørsmålet søker man å få et innblikk i hvordan den eksisterende arbeidskonteksten fungerer. Mye av denne informasjonen er allerede presentert i bakgrunnskapittelet. En forståelse av arbeidskonteksten er også, som introdusert i seksjon 3.1, helt essensielt for å ta i bruk CSCW til å skape suksessfulle datasystemer. Det første forskningsspørsmålet kan dermed karakteriseres som en inngangsportal til det andre spørsmålet der man spør hvordan CSCW kan benyttes til å støtte den eksisterende praksisen.

Det andre spørsmålet spør hvordan man, på bakgrunn av teori fra CSCW, kan støtte samarbeidende organisering av musikkmateriale gjennom et distribuert system. Formålet med systemet er å forenkle eller forbedre måtene som har vært i bruk tidligere. Å forenkle eller

forbedre innebærer i denne sammenhengen å minske eller fjerne problemene som er identifisert i analysen av intervjuene.

Gjennom det andre forskningsspørsmålet forsøker man likevel ikke å presentere en endelig løsning på hvordan CSCW *skal* brukes for å løse denne typen utfordringer; fokuset her er å introdusere en potensiell løsning som kanskje kan bidra med mer kunnskap innenfor CSCW-feltet. Funnene fra evalueringen vil for eksempel være med å skape et fundament for videre utvikling og forskning på akkurat denne typen forskningsspørsmål. Men kan man kalle dette for forskning hvis kjerneaktiviteten er å skape noe nytt for å løse et problem? I kapittel 6, om forskningstilnærming og metoder, skal det redegjøres for hvorfor denne typen problemløsning kan kalles forskning. I de neste to seksjonene vil jeg begrunne hva som skiller CMO i forhold til andre common information spaces, samt hvorfor CMO er utviklet som en klientapplikasjon i stedet for et webgrensesnitt.

## **5.1 CMO: et unikt Common Information Space?**

Tidligere ble Common Information Spaces (CIS) introdusert. Hva gjør imidlertid denne applikasjonen unik i forhold til andre CIS-systemer som tilbyr organiseringsfunksjonalitet over nettverket, samt muligheter for å interagere med andre brukere?

Det er riktig at CMO vil ha likheter med eksisterende CIS-systemer (eller shared workspaces). Men dette systemet tilbyr også eksplisitt støtte for avspilling av flere typer musikkfiler, samt integrerer muligheter for redigering og komponering av musikk i tredjehandsprogramvare. I tillegg er fremgangsmåten for organisering av musikkmateriale spesielt tilpasset komponister. Disse aktørene må ofte forholde seg til album, singler, settlist, og så videre. Hierarkioppbygningen av prosjekter (hoved- og underprosjekt) i CMO kan av den grunn oppfattes som naturlig av dem (se brukerkravene i seksjon 7.2.1). Applikasjonen har også støtte for både asynkron og synkron interaksjon, og noen musikere kan derfor møtes online uten at alle er tilstede for å laste inn nytt musikkmateriale, redigere, diskutere og planlegge hvordan komponeringen av et gitt musikkstykke skal forløpe seg fremover. Informasjonen om hva de har gjort vil videre være tilgjengelig for resten av deltakerne når de logger seg på systemet. Brukerinformasjon om hver bruker er også tilpasset de forskjellige rollene som eksisterer i

samarbeidet. Brukerinformasjonen presenterer informasjon om rolletype (musiker, lydtekniker, produsent) og eventuell informasjon om hvilket instrument en gitt bruker spiller, hvilket lydfirma han jobber i eller hvilket selskap han representerer (se seksjon 7.3.4 om brukerinformasjon). Summen av dette bidrar til å skille CMO fra et vanlig shared workspace.

Kunne man heller skreddersydd et eksisterende CIS for å tilby den samme funksjonaliteten? Det er mulig at man kunne benyttet et samarbeidsverktøy som Microsoft Sharepoint<sup>15</sup> eller tilsvarende som et utgangspunkt for å tilby den samme funksjonaliteten. Men dette ville uansett krevd kunnskap og innsikt i systemarkitekturen og all funksjonaliteten som eksisterer her, og det er ikke nødvendigvis en enkel oppgave. I tillegg vil det være kostnader involvert ved å anskaffe eksempelvis Microsoft Sharepoint; noe som ikke er aktuelt når man utvikler programmet selv i et gratis utviklingsmiljø (se seksjon 7.1 for mer informasjon om valg av teknologi og språk).

## **5.2 Klientapplikasjon versus webgrensesnitt**

For å besvare det andre forskningsspørsmålet har det blitt utviklet en applikasjon. Det finnes flere fremgangsmåter for utvikling av en applikasjon, men i dette studiet har CMO har blitt designet som en lokal klientapplikasjon som kommuniserer med en server. Hvorfor har ikke CMO blitt designet som et webgrensesnitt med den samme funksjonaliteten?

En del av funksjonaliteten kan uproblematisk implementeres i et webgrensesnitt, men det finnes likevel en del praktiske årsaker til at en lokal klientapplikasjon passer bedre i dette tilfellet: (1) Filredigering av for eksempel midi, mp3-er, wave-filer, Pro Tools-filer<sup>16</sup>, GarageBand-filer<sup>17</sup> er ikke særlig utbredt på web. Lydfiler har en formidabel størrelse (flere megabytes), og denne typen filredigering krever mye prosesseringskraft. Om dette var mulig å gjøre fra bare serverens side, så ville det lagt mye last på den. (2) En webside har i utgangspunktet ikke tilgang til å manipulere lokale fillagre av sikkerhetshensyn. Denne typen begrensninger er pålagt flere nettlesere (Andersen & Abella, 2004; Mozilla, 2008). (3) Ved å ha et lokalt klientfillager som

---

<sup>15</sup> Microsoft Sharepoint: <http://www.microsoft.com/sharepoint/default.aspx>

<sup>16</sup> Pro Tools: <http://www.protools.com>

<sup>17</sup> Garage Band: <http://www.apple.com/ilife/garageband/>

synkroniseres opp mot serverens fillager, kan man raskt spille av ønskede filer uten å måtte bruke direkteavspillingsfunksjonalitet for web. Direkteavspilling på web krever at man laster inn filen fra serveren hver gang den skal spilles av (med mindre nettleseren nettopp har lagt filen til i den lokale hurtigbufferen), og dette krever mer tid og båndbredde enn å laste ned filen en gang på klienten. (5) Dersom all organisering, redigering og interaksjon skulle foregått over web, ville man vært avhengig av kontinuerlig høye bredbåndshastigheter og en meget rask server. Lave overføringshastigheter ville gitt dårlig brukskvalitet til brukerne. Ved å kjøre applikasjonen gjennom et klient-/serverforhold, der klienten har ansvar for prosessering, er man til gjengjeld ikke avhengig av dette.

I tillegg blir det antatt at det ikke er problematisk for komponister å måtte installere en lite krevende applikasjon fremfor kun å bruke web hvis applikasjonen kan tilby dem et bedre og mer effektivt arbeidsverktøy. I bakgrunnskapitlet ble det vist at alle intervjuobjektene benyttet flere IKT-applikasjoner fra før, og dette er av den grunn ikke ukjent for dem.

I kapittel 7 vil jeg gå nærmere inn på designvalgene for systemet med basis i intervjuene og teorien som ble gjennomgått tidligere, men først skal det redegjøres for valg av forskningstilnærming og metoder.

## 6. Forskningstilnærming og metodevalg

Dette kapittelet vil handle om *designforskning*, valg av datainnsamlingsmetode og valg av evalueringsmetode for den ferdigutviklede løsningen. I tillegg skal det vises hvordan disse delene er relatert til hverandre. Formålet med dette kapittelet er å illustrere hvilke fremgangsmåter og perspektiver som er valgt for å besvare forskningsspørsmålene i studiet. I forrige kapittel aktualiserte jeg en problemstilling: hvordan kan innovativ og skapende problemløsning falle innenfor forskningens domene? Første seksjon skal behandle nettopp dette temaet.

### 6.1 Designforskning

Forskning kan generelt defineres som en aktivitet som bidrar til forståelse av et fenomen. I mange tilfeller forekommer fenomenene naturlig, enten som del av natur eller samfunn. Forskerens rolle er ofte å forstå fenomenene uten å påvirke dem direkte. Designforskning<sup>18</sup> er, i følge Association for Information Systems (AIS), en tilnærming som tvert i mot baserer seg på forskerens påvirkning av fenomenene som skal undersøkes (2007). I følge AIS handler designforskning om:

«[...] the analysis of the use and performance of designed artifacts to understand, explain and very frequently to improve on the behavior of aspects of Information Systems. Such artifacts include - but certainly are not limited to - algorithms (e.g. for information retrieval), human/computer interfaces and system design methodologies or languages» (2007).

Her ser man at forskerens rolle er å designe et artefakt, for så å forstå og eventuelt forbedre et eller annet aspekt ved det. Simon (1982, pp. 5-8) drar et tydelig skille mellom *sciences of the artificial* (et av utgangspunktene for designforskning) og *natural sciences*. Han beskriver natural sciences som «a body of knowledge about some class of things – objects or phenomena – in the world: about the characteristics and properties they have; about how they behave and interact with each other». Sciences of the artificial omfatter på den annen side menneskeskapte fenomener eller objekter som er fremstilt for å tilfredsstille noen forhåndsbestemte mål. Dette forholdet kan i følge Simon (*ibid*) kalles en dikotomi, der man skiller mellom det *deskriptive*, hvordan noe er, og det *normative*, hvordan noe bør være. Inndelingen er ikke plassert innenfor noen spesiell forskningstradisjon, det være seg naturvitenskap eller samfunnsvitenskap; dikotomien er et

<sup>18</sup> Eng.: Design research el. design science

forsøk på å skape et skille mellom (i) det menneskeskapte og hvordan det bør være (for å nå noen predefinerte mål), og (ii) det naturlige og hvordan det er. Men samtidig kan man i følge Simon anvende deskriptive metoder fra natural sciences for å forstå det menneskeskapte artefaktet.

På veien mot forståelse må man imidlertid først designe det som skal forstås. Hva er design? Begrepet har blitt definert som «the use of scientific principles, technical information and imagination in the definition of a structure, machine or system to perform pre-specified functions with maximum economy and efficiency» (Fielden 1975, hentet fra Walls, Widmeyer, & El Sawy, 1992, pp. 36-37). Walls et al. (1992, p. 42) har videre skildret begrepet som bestående av et *produkt* og en *prosess*. Produktet er artefaktet som skal bli realisert. Prosessen er selve planleggingen av for eksempel maskindelene eller strukturene slik at alle kravene til artefaktet blir tilfredsstilt. Isolert sett er begrepet *design* tydelig normativt; man har en plan om hvordan et objekt eller fenomen bør designes. For at design kan bli del av forskning, må det imidlertid være deskriptive aktiviteter tilstede, der målet er å forstå hvordan artefaktet er eller fungerer. Undersøking, testing og evaluering av et artefakt er eksempler på slike deskriptive aktiviteter. Walls et al. hevder i denne sammenhengen at en designer kan kalles en forsker hvis han konstruerer redskaper for å teste teorier (1992, p. 38). I dette studiet blir designet av applikasjonen den normative aktiviteten, og evalueringen av applikasjonen utgjør den deskriptive aktiviteten. Bakgrunnen for hvordan applikasjonen *bør* designes, er basert på tidligere forskning og intervjuer med aktører innenfor musikkbransjen. Gjennom evalueringen av applikasjonen søker man å forstå hvordan applikasjonen *er*, mer eksplisitt måle hvordan applikasjonen fungerer (gjennom et sett av måleegenskaper) når flere komponister anvender den.

### **6.1.1 Designforskning på informasjonssystemer**

Hevner et al. beskriver to distinktive forskningsparadigmer innenfor forskningen på informasjonssystemer (IS) som kan illustrere designets innflytelse på forskning (Hevner, March, Park, & Ram, 2004). Paradigmene kalles designforskningsparadigmet og atferdsforskningsparadigmet. *Atferdsforskning*<sup>19</sup> søker å forstå eller forutsi organisatoriske eller menneskelige fenomener innenfor analyse, design, implementering, ledelse og bruk av IS, og

---

<sup>19</sup> Eng.: Behavioural science



dette paradigmet er opptatt av *sannferdige* fenomensforklaringer (i hvert fall i den grad forskningsbidrag er sannferdig). Designforskning er, på den annen side, fokusert på *nytteredskaper*<sup>20</sup>. Men disse paradigmene er ikke isolert fra hverandre; sannferdige fenomensforklaringer gir informasjon om hvordan man best kan designe nytteredskaper, og evalueringen av nytteredskaper bidrar til sannferdig teori (og fenomensforklaringer) som kan videreutvikles og forbedres (Hevner et al., 2004, pp. 76-80).

Hva er grunnene til å velge designforskning som tilnærming i dette studiet? Schmidt & Bannon (1992) hevder at CSCW er et designorientert forskningsfelt, og de mener at design av eksperimentelle CSCW-systemer kan gi viktig innsikt i karakteristikken og kravene til samarbeid (pp. 11, 12). Designforskning som tilnærming passer derfor godt i sammenhenger der CSCW utgjør det teoretiske rammeverket. Designforskning kan i tillegg betraktes som en tilnærming rettet mot problemløsning. Ofte er utgangspunktet at man har identifisert et problem innenfor en organisatorisk eller sosial arena (Hevner et al., 2004, p. 77). I dette studiet har det tidligere blitt presentert en kontekst med problemer og utfordringer. For å løse disse utfordringene er design av et artefakt en av flere mulige fremgangsmåter. Oppbygningen av forskningsspørsmålene harmonerer med dette tenkesettet. Det første forskningsspørsmålet søker å forstå hvordan en aktivitet forløper seg i eksisterende praksis, og identifiserte problemer her. Det andre forskningsspørsmålet spør: hvordan kan man forbedre eller forenkle det eksisterende samarbeidet? Og her blir det først aktuelt å snakke om design. En måte å forbedre det eksisterende samarbeidet er nettopp ved bruk av et artefakt. Artefaktet blir da svaret på dette ”hvordan”. Men artefaktet svarer ikke for seg selv om det har vært med å forenkle eller forbedre. For å få svar på dette, må man, som nevnt, teste et eller flere aspekter av artefaktet gjennom en evaluering. Valg av evalueringsmetode blir drøftet senere.

Hva er forskjellene mellom designforskning og alminnelig design/utvikling av et informasjonssystem? Alminnelig design og utvikling av et system går ut på å benytte eksisterende kunnskap til å løse et organisatorisk problem, for eksempel gjennom å utvikle et økonomisystem for å oppnå kontroll over de økonomiske aspektene i en organisasjon.

---

<sup>20</sup> Eng.: Utility

Kunnskapen som blir brukt i et slikt eksempel vil allerede være en del av *IS-kunnskapsbasen*<sup>21</sup> og allment kjent innenfor IS-forskning. IS-kunnskapsbasen består av fundamenter, metodologier, og frukter av tidligere IS-forskning, for eksempel teorier, rammeverk, verktøy, metoder, modeller, etc. Designforskning streber etter, i dette henseende, å tilføye noe nytt her, eller med andre ord, å løse ubesvarte problemstillinger på en innovativ eller mer effektiv måte, og på denne måten etterlate et kunnskapsbidrag til IS-kunnskapsbasen (Hevner et al., 2004, pp. 80-81). Hovedforskjellen ligger altså i forholdet mellom å bruke kunnskap fra IS-kunnskapsbasen og å tilføye kunnskap til IS-kunnskapsbasen.

Hvordan kan dette studiet relateres til designforskning? Ettersom distribuert samarbeidende organisering av musikkmateriale ikke har hatt veldig mye i fokus i en CSCW-sammenheng, vil utviklingen og evalueringen av dette artefaktet kunne bidra med informasjon til IS-kunnskapsbasen, og dette artefaktet (og dets evaluering) kan senere utgjøre fundamentet for videreutvikling gjennom en ny omgang designforskning eller atferdsforskning.

## 6.2 Kvalitative metoder

For å forstå hvordan musikkorganisering kan fasiliteres gjennom en CSCW-applikasjon, har det blitt benyttet en kvalitativ datainnsamlingsmetode. Denne metoden har bidratt til å svare på det første forskningsspørsmålet, det vil si forståelse for eksisterende praksis, men også bidratt med informasjon til kravspesifikasjonen i dette studiet. I bakgrunnskapittelet ble flere av funnene fra intervjuene presentert.

Kvalitativ data kan bli definert som informasjon uttrykt gjennom ord eller bilder. Kvantitativ data er informasjon uttrykt gjennom numre eller diskrete kategorier (Seaman, 1999, p. 7). Som nevnt tidligere, søker CSCW å forstå hvordan samarbeid fungerer, og derav designe datamaskinbaserte teknologier til å støtte dette. Hovedgrunnen til å benytte kvalitativ datainnsamling er at området som skal undersøkes er sentrert rundt menneskelig atferd. Problemområdet er komplekst, og det er mange aktører å forholde seg til. Samarbeid kompliserer

---

<sup>21</sup> Knowledge Base

situasjonen ytterligere siden det eksisterer en gjensidig avhengighet mellom aktørene som er involvert i samarbeidet. Når det eksisterer et rikt avhengighetsforhold mellom flere brukere, kan det være lett å overse enkelte aspekter dersom man eksempelvis hadde tatt utgangspunkt i å bruke kvantitative spørreskjemaer. Seaman (*ibid*) antyder at menneskelig atferd (relatert til motivasjon, kommunikasjon og forståelse) ofte krever kvalitative metoder til datainnsamlingen fordi kvantitative metoder har hatt evne til å abstrahere bort kompleksiteten av atferden. Grimen (2001, p. 200) nevner at kvalitative metoder er et godt utgangspunkt til bruk i forstudier. Intervjuene kan i den sammenhengen betraktes som del av forstudiet for å finne ut om det finnes problemer eller utfordringer som kan bedres gjennom CSCW. Uten en forstudie og en informasjonsbakgrunn om den eksisterende arbeidskonteksten, er det vanskelig å vite om det finnes områder som kan være gjenstand for forbedring. Med informasjonsbakgrunn menes det at undertegnede har erfaring fra musikkpraksis og samarbeidende komponering. Dette omtales nærmere i seksjon 9.1 om evaluering av studiet.

I begynnelsen av studiet ble det utført 7 ustrukturerte intervjuer. Ustrukturerte intervjuer går ut på å utforske et område sammen med intervjuobjektet, uten å ha en fastsatt ramme for rekkefølgen eller typen spørsmål som stilles (Robson, 2002, pp. 278-281). Fordelen med kvalitative intervjuer i dette studiet er at de er godt skikket til utforsking av en arbeidskontekst, samt avdekke komplekse forhold som eksisterer her. Ulempen med kvalitative intervjuer er at de ikke kan kvantifiseres, og det er vanskelig å gjøre generaliseringer eller statistiske bemerkninger på bakgrunn av dem. Men samtidig har det ikke intensjonen vært å kunne foreta eksplisitte generaliseringer om hvordan samarbeidende musikkorganisering foregår i større populasjoner. Bakgrunnen for å utføre intervjuene har, som nevnt, vært basert på ønsket om å finne ut hvorvidt det eksisterer et problem innenfor arbeidskonteksten som undersøkes, samt å utforske denne arbeidskonteksten for å samle inn empirisk data. Funnene fra intervjuene har blitt analysert i den grad at man har funnet enkelte tendenser, men disse funnene må tolkes som veiledende.

Intervjuobjektene er alle bekjente av undertegnede. Jeg har vært involvert i musikk samarbeid med de fleste av dem tidligere. Intervjuene ble både utført ansikt til ansikt på forskjellige kafeer i Bergen sentrum og via e-post og telefon. (e-post og telefon ble benyttet dersom det var vanskelig å utføre intervjuene på grunn av manglende tid til oppmøte fra deres side eller dersom

intervjuobjektene befant seg i andre byer). Fokuset i intervjuene var å finne ut om deres erfaringer med samarbeidende musikkproduksjon og hvordan de oppbevarte, overleverte og kommuniserte om musikkmaterialet under komposisjonsprosessen. Intervjuguide og funn er vist i appendiks B.

### 6.3 Evaluering: Groupware Walkthrough

Evalueringen av applikasjonen vil bli utført etter prinsippene definert i Groupware Walkthrough<sup>22</sup> (GWW). Evaluering er et helt essensielt aspekt innenfor designforskning, og det mest vesentlige her er å finne formålstjenlige måleegenskaper ved designet (Hevner et al., 2004, p. 85). Med dette menes det at resultatet skal kunne måles etter definerte kriterier, og måleresultatene skal kunne overføres til IS-kunnskapsbasen for potensiell videreutvikling og forbedring på et senere tidspunkt. GWW er tilrettelagt av Pinelle & Gutwin (2002), og er basert på kognitiv gjennomgang<sup>23</sup>. Målet med metoden er å inspisere brukskvaliteten<sup>24</sup> til en gruppevareapplikasjon. Brukskvaliteten blir da den egenskapen som skal måles i denne sammenhengen. GWW skiller seg fra kognitiv gjennomgang (som bare er fokusert på enkeltbrukerprogramvare) ved å rette fokus på kompleksiteten i flerbrukerprogramvare. I denne sammenhengen har Gutwin og Greenberg (1999, p. 247) definert groupware-brukskvalitet som: «...the degree to which a groupware system supports the mechanics of collaboration for a particular set of users and a particular set of tasks». Tidligere ble samarbeidsmekanismene til Gutwin og Greenberg (2000) gjennomgått, og disse mekanismene blir også anbefalt til bruk i GWW av Pinelle og Gutwin (2002, pp. 456-455). Fremgangsmåten for måling av samarbeidsmekanismene blir gjennomgått i neste seksjon.

GWW består av to hovedkomponenter: en Group Task Model<sup>25</sup> (GTM) for å identifisere og modellere det virkelige arbeidet som finner sted, og en Walkthrough Process<sup>26</sup> (WP) av systemet for å evaluere systemets grensesnitt (Pinelle & Gutwin, 2002, p. 455). Forholdet mellom GTM og WP er illustrert i figur 3:

---

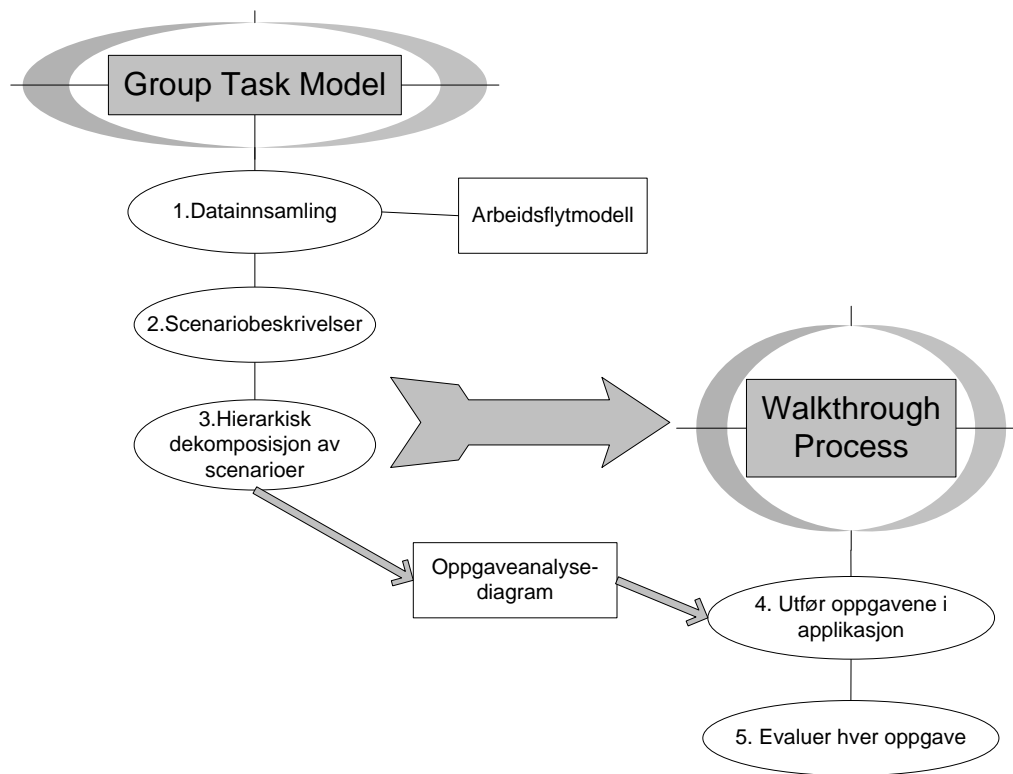
<sup>22</sup> Nor.: gruppevaregjennomgang

<sup>23</sup> Eng.: Cognitive Walkthrough

<sup>24</sup> Eng.: Usability

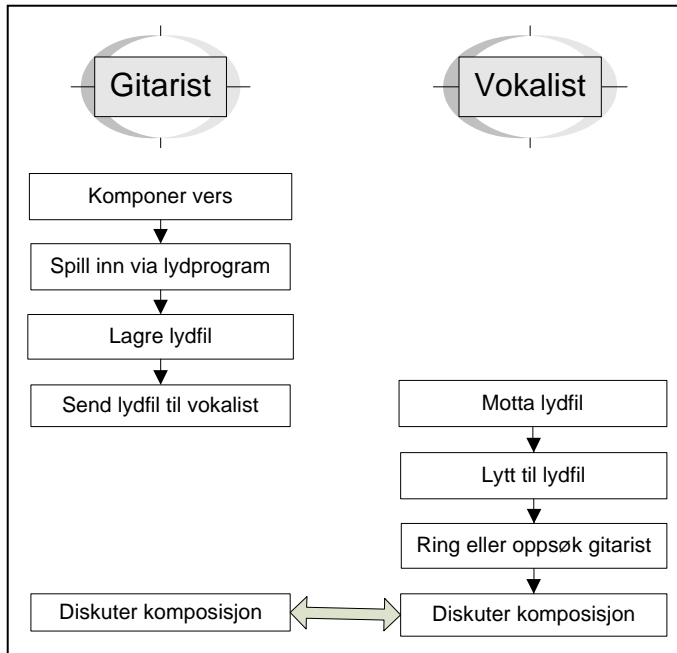
<sup>25</sup> Nor.: Gruppebasert oppgavemodell

<sup>26</sup> Nor.: Gjennomgangsprosess



**Figur 3 – Komponentene i Groupware Walkthrough.**

GTM består i hovedsak av tre analyseaktiviteter. Den første aktiviteten innebærer datainnsamling, og har blitt utført gjennom kvalitative intervjuer med komponister. Målet med aktiviteten er å kunne skille mellom samarbeid og individuelt arbeid fra den eksisterende konteksten. En arbeidsflytmodell kan være hensiktsmessig for å identifisere hvor samarbeidsaktivitetene foregår mellom to eller flere brukere. Individuelle aktiviteter er ikke interessant i denne evalueringsmetoden. Disse kan evalueres ved hjelp av eksisterende rammeverk for enkeltbrukerapplikasjoner (Pinelle & Gutwin, 2002, p. 457). Se figur 4 for et eksempel på en arbeidsflytmodell av aktivitetene fra to komponister sitt arbeid.



**Figur 4 – Arbeidsflytmodell.**

Fra eksemplet kan man identifisere et sted der samarbeid foregår. Dette er uttrykt med pil mellom gitaristen og vokalist, og forekommer der aktørene diskuterer komposisjonen.

Samarbeid utgjør grunnlaget for den andre aktiviteten i GTM: scenariobeskrivelser. Et scenario er en deskriptiv formalisering av det generelle

arbeidet som brukerne utfører, hvem som inngår i arbeidet og hvilken kunnskap de har. Scenarioene bør inneholde eksplisitte brukerbeskrivelser, beskrivelse av tilsiktet utfall og beskrivelse av omstendighetene (Pinelle & Gutwin, 2001, p. 105). Disse tilleggsmomentene gir viktig kontekstuell informasjon når GWW blir utført.

Selv om man fokuserer på samarbeid, må man likevel ikke neglisjere de individuelle handlingene som ledet til samarbeidet. I tabell 3 er det gjengitt et eksempel på en scenariobeskrivelse basert på samarbeidet som ble identifisert i arbeidsflytmodellen. B#1 og B#2 uttrykker forskjellige brukerroller, og i følgende scenario er brukerne henholdsvis en gitarist og en vokalist.

Tabell 3 – Scenariobeskrivelse. (Pinelle &amp; Gutwin, 2002).

**Scenario: Diskusjon om komposisjon**

<b>Aktivitetsbeskrivelse:</b>	B#1 (gitarist) ønsker å diskutere en komposisjon med B#2 (vokalist). Først må han få tak i B#2, og forhøre seg hvorvidt B#2 har tid til en diskusjon. Etterpå må han spesifisere hvilken komposisjon han ønsker å diskutere, samt forsikre seg om at B#2 er kjent med komposisjonen.
<b>Brukerspesifikasjon:</b>	Gitaristen har stort sett ansvaret for å lage grovskissene til låtene. Han komponerer med gitar og spiller inn komposisjonene i et lydprogram. Vokalistene skriver tekster og har stor innflytelse på melodilinjene til vokalstemmen. Han må blant annet passe på at vokallinjene er tilpasset hans stemmeregister.
<b>Tilsiktet utfall og resultat:</b>	Utteksle meninger om komposisjonen.
<b>Forhold/omstendigheter:</b>	Vokalistene og gitaristen arbeider fra forskjellige lokasjoner. De bruker e-post til både asynkron kommunikasjon og formidling av komposisjoner. Telefon blir benyttet til synkron kommunikasjon.

Scenarioene utgjør grunnlaget for den tredje analyseaktiviteten i GTM. Aktiviteten kalles hierarkisk dekomposisjon og går ut på å dekomponere scenarioene til oppgaver. Oppgavene blir beskrevet gjennom erklæringer som uttrykker hva som skjer i scenarioet, og disse deles igjen opp i individuelle underoppgaver og samarbeidende underoppgaver.

Hierarkisk inndeling av oppgavene (Pinelle & Gutwin, 2002, p. 457)

## 1. Scenario

### 1.1. Hovedoppgave

#### 1.1.1. Individuell underoppgave

#### 1.1.2. Samarbeidende underoppgave.

Forholdet mellom hovedoppgaver og underoppgaver kan beskrives slik: hovedoppgavene beskriver *hva* som skjer i et scenario, mens underoppgavene beskriver *hvordan* en aktivitet blir utført. Dekomposisjonen er den siste analyseaktiviteten i GTM, og denne aktiviteten utgjør



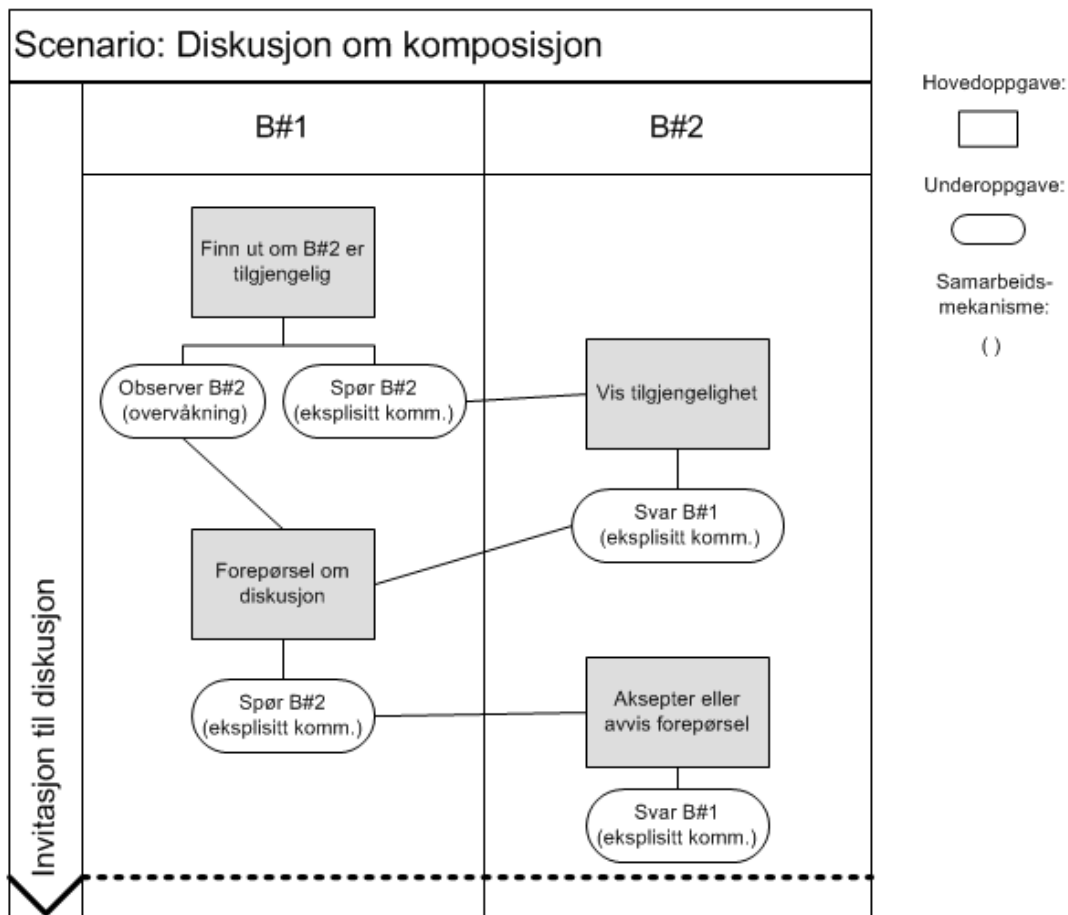
grunnlaget for modellering av oppgavene til bruk i analysediagrammer<sup>27</sup>. Analysediagrammene viser forskjellige fremgangsmåter som skal kunne benyttes for å løse en gitt oppgave i applikasjonen, og analysediagrammene blir anvendt under WP (det vil si gjennomgangsproessen). Scenariobeskrivelsen er en beskrivelse av hvordan ting ble utført i eksisterende praksis; analysediagrammene viser en oversikt over hvordan denne aktiviteten kan løses gjennom et sett av oppgaver som er tilpasset applikasjonen som er gjenstand for evaluering. WP blir iverksatt ved at en vurderingsansvarlig<sup>28</sup> går gjennom hver oppgave i analysediagrammene sammen med potensielle brukere av systemet og ser i hvilken grad applikasjonen støtter gruppemedlemmene i å nå et bestemt mål. Her er det viktig å bemerke seg at alternative løsningsmetoder absolutt er mulig, og at alle resultater må registreres. I dette studiet har evalueringen blitt gjennomgått i to omganger med to forskjellige brukere. Antallet brukere ble begrenset fordi jo flere brukere som er involvert, dess flere vurderingsansvarlige må være involvert. Pinelle & Gutwin (2002) nevner at flere brukere kompliserer evalueringen og overvåkingen av samarbeidsmekanismene, og at få brukere kan være tilstrekkelig for å avdekke mangler på brukskvaliteten (p. 461). I tillegg nevner de at en evalueringsansvarlig kan påta seg rollen som bruker og gå gjennom oppgavene i evalueringen sammen med en annen bruker (Pinelle & Gutwin, 2002, p. 459). Sistnevnte har også vært tilfelle i dette studiets evaluering, men i to separate gjennomganger.

Et eksempel på bruk av analysediagrammer blir illustrert i figur 5. Dette diagrammet viser oppgaver som ble identifisert gjennom den hierarkiske dekomposisjonen av scenariobeskrivelsen i forrige eksempel. Scenarioet skildret diskusjonen mellom vokalisten og gitaristen. Hovedoppgavene illustreres i firkanter og underoppgavene vises i ovale rektangler. Hver underoppgave samsvarer med en av samarbeidsmekanismene til Gutwin & Greenberg (2000). Dette er uttrykt i parentes.

---

<sup>27</sup> Eng.: Task analysis diagram

<sup>28</sup> Eng.: Evaluator



Figur 5 – Analysediagram av scenarioet diskusjon om komposisjon

Ovenfor ser man et scenarioutdrag som har blitt organisert i en naturlig sammenhengende sekvens. Sekvensen kalles «invitasjon til diskusjon», og inntreffer før gjennomføringen av diskusjonen kan ta sted. Analysediagrammet viser to forskjellige fremgangsmåter for å finne ut om gitaristen er tilgjengelig. Vokalisten kan enten observere gitaristen i systemet eller spørre ham om han er tilgjengelig. Dette analysediagrammet blir benyttet under evalueringen av systemet og får en grundigere gjennomgang der. I seksjon 7.2.4 blir omstendighetene rundt gjennomføringen av GWW presentert. Dette innebærer en presentasjon av brukerne, forskerens rolle og hvordan GWW ble gjennomført.

### 6.3.1 Målekriterier for brukskvalitet

Samarbeidsmekanismene representerer basisaktiviteter i et shared workspace. For å måle hvorvidt disse gjenspeiler god brukskvalitet, må man definere noen målekriterier.

I denne sammenhengen har Gutwin og Greenberg (2000) introdusert tre kriterier<sup>29</sup> som bør vurderes etter evalueringen av et gruppevare-system (p. 100):

- *Effectiveness* innebærer om en gitt oppgave ble vellykket utført, og her må man ta hensyn til antallet og omfanget av feil som inntraff i løpet av aktiviteten. Jo mindre feil, dess mer effectiveness.
- *Efficiency* måler ressursene (tid og anstrengelse) som er krevd av brukerne for å utføre en oppgave. Et godt gruppevaresystem muliggjør at samarbeidsaktivitetene kan utføres med mindre tid og anstrengelse enn et system med dårlig brukskvalitet.
- *Satisfaction* omhandler hvorvidt deltakerne var fornøyd med prosessene og utfallet av hver av oppgavene som inngikk i samarbeidet. Dette punktet kan overlappe med effectiveness og efficiency.

Resultatet av GWW kan brukes iterativt til faktiske forbedringer av kravspesifikasjonen, designet eller koden, men dette er utenfor rekkevidden og tiden til dette studiet. Resultatene fra evalueringen vil på den annen side bli registrert og dokumentert slik at eventuelle forbedringer av svakheter eller effektiviseringer kan gjennomføres i en etterfølgende studie. Noen bugs ble imidlertid rettet mellom den første og den andre evalueringsgjennomgangen.

I dette kapittelet har forskningstilnærming og valg av innsamlings- og evalueringsmetoder blitt presentert. Intensjonen med dette kapittelet var å beskrive fremgangsmåtene for å løse forskningsspørsmålene. I de neste to kapitlene skal jeg presentere designet og evalueringen av CMO.

---

<sup>29</sup> Nor.: Effektivitet, nyttevirkning/virkningsgrad, tilfredsstillelse.

## 7. Design av Collaborative Music Organizer

I dette kapitlet vil jeg rette fokus mot designet av applikasjonen: Hvorfor ser applikasjonen ut slik den ser ut? Hvilken funksjonalitet støtter den og hvorfor? Hvordan har applikasjonen blitt utviklet? Dette er spørsmål som får oppmerksomhet her. Mer eksplisitt vil jeg vurdere valg av teknologi, klient-/serverarkitektur, brukerkrav, fremgangsmåte for utviklingen, og begrunne designvalg med funn fra datainnsamlingen, samt teorien som ble gjennomgått i kapittel 3 om teoretisk rammeverk. I tillegg vil det bli foretatt en klassifisering av CMO i henhold til Weinbergs (2002) klassifiseringssystem.

I presentasjonen av designvalgene vises det hvordan teori fra CSCW har blitt brukt under designprosessen for å besvare forskningsspørsmålet om hvordan man kan støtte samarbeidende organisering i eksisterende praksis. Se appendiks A for en nettlink til en videopresentasjon av CMO i praksis.

### 7.1 Valg av teknologi og språk

Basisen for teknologi i dette studiet har vært Microsoft sitt .NET-rammeverk<sup>30</sup>. Dette rammeverket tilbyr et rikt utviklingsmiljø der man kan bruke mange forskjellige programmeringsspråk, bibliotek og databasesystemer sammen. I utgangspunktet var .NET ment for Windows-plattformene, men ved å bruke den åpne kildekoden Mono<sup>31</sup> kan man kjøre .NET-applikasjoner på Linux, Solaris, Mac og Unix. C# er navnet på programmeringsspråket som har blitt benyttet til applikasjonsutviklingen. Det er flere grunner til å velge C# og .NET. For det første er det veldig lett å integrere med tilgjengelige biblioteker i Windows, samt lett å integrere med andre programmeringsspråk (for eksempel i en annen forskningssammenheng). Og for det andre er det gratis å laste ned. SQL Server 2005 Express<sup>32</sup> har blitt benyttet som databasehåndteringssystem (DBMS), og dette systemet er også gratis.

---

<sup>30</sup> Overview of .NET: <http://msdn2.microsoft.com/en-us/netframework/aa569294.aspx>

<sup>31</sup> What is Mono? [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)

<sup>32</sup> Sql Server 2005 Express Edition: <http://www.microsoft.com/sql/editions/express/default.msp>

Serveren er satt opp med operativsystemet Windows Server 2003, og på denne finnes følgende 3 teknologier som er essensielle for å tilby interaksjon og koordinasjon mellom brukerne:

- 1) DBMS
- 2) ChatServer-program <sup>33</sup> (som håndterer tilkoblede klienter og delegering av direkte meldinger mellom dem).
- 3) FTP-område (som utgjør lagringsplassen for synkronisering av dokumenter mellom brukerne).

Språket i applikasjonen er lagt opp på engelsk. Dette er gjort bevisst fordi nesten samtlige programmer innenfor musikkproduksjon bruker engelsk terminologi. Dersom man hadde anvendt norsk, måtte man oversatt de engelske begrepene til norsk, og det er ikke sikkert at denne oversettelsen og meningen av ordene ville vært intuitivt for brukerne. I tillegg har det blitt nevnt at tredjehåndsapplikasjoner skal kunne benyttes for å redigere dokumenter i CMO, det vil si at en gitt fil skal kunne åpnes og endres i et annet program via CMO. Dersom CMO hadde hatt basis i norsk språk, kunne det muligens oppstått forvirring og frustrasjon hos brukerne hvis de måtte forholde seg til både norsk- og engelskspråklige applikasjoner.

### **7.1.1 Distribuert organisering via klient-/serverarkitektur**

I seksjon 2.1 ble det redegjort for hva distribuert organisering betyr. Hvordan har dette blitt realisert i forhold til IKT-løsningens systemarkitektur? I dette studiet har distribuert organisering blitt gjennomført ved at flere klienter kommuniserer med en sentralisert server gjennom et klient-/serverforhold. Klient-/serverarkitekturen har vært basert på en *fat klient-modell*<sup>34</sup> der klienten tar seg av applikasjonsprosesseringen, og serveren håndterer datalagringen (Sommerville, 2007, pp. 270-275). Denne arkitekturen muliggjør at applikasjonen kan koordinere organisere, oppbevare og oppdatere musikk materialet med serveren, mens serveren opprettholder en koherent systematisering av det lagrede musikk materialet. Men i tillegg er klienten avhengig av å kunne lagre musikk materialet lokalt for redigering eller avspilling, og når klienten startes vil alt

---

<sup>33</sup> Se skjermbilde i appendiks F

<sup>34</sup> Eng.: Fat-client model

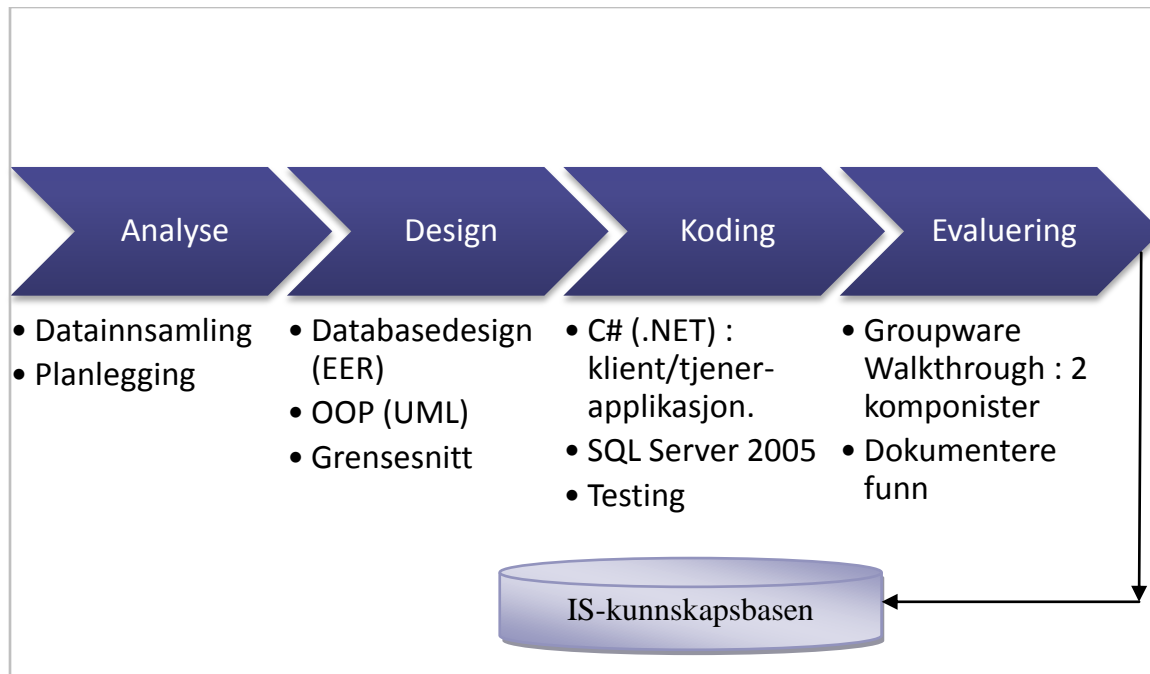
relevant musikkmateriale automatisk synkroniseres med det musikk materialet som finnes på serveren.

## 7.2 Prosessmodell

Den praktiske fremgangsmåten for programvareutviklingen i dette studiet har vært basert på en Software Process Model <sup>35</sup> (SPM). Med SPM menes aktivitetene som resulterer i et programvareprodukt. En SPM er en forenklet og abstrahert beskrivelse av utviklingsprosessen som fremstiller et avgrenset perspektiv på disse aktivitetene (Sommerville, 2007, pp. 8-9;65). Perspektivet i en slik modell er avgrenset fordi aktivitetene som inngår i avgrensningen ansees å være tilstrekkelig for å realisere et ønsket informasjonssystem på best mulig måte i henhold til tid, ressurser, prioriterte områder og målsetninger. Utgangspunktet for SPM i dette studiet er den tradisjonelle programvareutviklingssyklusen basert på fossefallsmodellen (Sommerville, 2007, pp. 66-68), men denne syklusen ble videre avgrenset til kun å omhandle analyse, design, koding og evaluering. Vedlikehold, sikkerhetstiltak, opplæring og brukerdokumentasjon har ikke vært tatt hensyn til. Denne avgrensningen var nødvendig (og langt fra ulovlig) siden dette er en studie der *en* person er ansvarlig for utviklingen av systemet. Kodingsfasen tilrettela ikke for iterative programmeringssykluser basert på kontinuerlige tilbakemeldinger fra testbrukere. Iterasjon med brukerdeltakelse ville vært ønskelig dersom det hadde vært mer tid eller hvis applikasjonen skulle settes ut i produksjon.

---

<sup>35</sup> Nor.: Programvareprosessmodell



**Figur 6 – Software Process Model for utviklingen av Collaborative Music Organizer.**

Modellen i figur 6 illustrerer hovedstegene i syklusen. Prosessmodellen har også ekspandert i dette studiet; på figuren ser man at det er en sammenkobling mellom evalueringen og IS-kunnskapsbasen slik denne sammenhengen fremkommer i designforskning. Evalueringen vil, som nevnt, bidra med ny informasjon til kunnskapsbasen, på samme måte som tidligere IS-forskning og resultater fra nærliggende disipliner har bidratt (Hevner et al., 2004, pp. 80-81). Men dokumentasjonen av designvalgene bak CMO vil også være et betydningsfullt bidrag til IS-kunnskapsbasen. Hvert av hovedstegenes innhold i modellen blir utdypet i de følgende seksjonene.

### 7.2.1 Analyse

Den første fasen i prosessmodellen har vært preget av datainnsamling og planlegging. De kvalitative intervjuene har bidratt med informasjon om både arbeidskonteksten som undersøkes, samt hvilke krav man bør stille til applikasjonen. Forskningslitteratur har også vært en viktig del i prosessmodellens analysefase. Samlet bidro disse momentene til at det ble konstruert en kravspesifikasjon for systemet. Innenfor softwareutvikling kan man benytte to typer kravspesifikasjoner (Sommerville, 2007, pp. 118-127). Den første kalles brukerkrav<sup>36</sup> og er en

<sup>36</sup> Eng.: User Requirements

høynivå abstrakt oversikt som viser hva systemet forventes å kunne gjøre eller hvilken funksjonalitet det skal kunne tilby. Denne oversikten viser kun den eksterne oppførselen til systemet og bryr seg ikke om designkarakteristikkene. Den andre inndelingen kalles systemkrav og er en lavnivå beskrivelse av funksjonene, tjenestene og driftbegrensningene av systemet. Sistnevnte er relativt detaljert og er i hovedsak forbeholdt aktørene som inngår i utviklingen av systemet, eksempelvis systemarkitekter og programmerere. For å forenkle fremstillingen av kravspesifikasjonen vil derfor brukerkravene settes i fokus her. En presentasjon av brukerkravene til applikasjonen er anvist i følgende tabell:

Tabell 4 – Brukerkravene til Collaborative Music Organizer.

Krav	Beskrivelse av krav
1. Innlogging	Alle brukere må logge inn med eget brukernavn og passord. På innloggingsskjermen velger de også hvilket sted (hjemme/studio/etc.) de befinner seg på når de logger inn.
2. Prosjektutforsker	<p>Programmet skal tilby et utforskergrensesnitt med oversikt over prosjektene og dokumentene som er opprettet av brukerens gruppe. Brukeren skal selv kunne bestemme hvor og hva han skal utforske. Grensesnittet til prosjektutforskeren skal baseres på et hierarki som er velkjent for komponister. Hierarkiet er bygget opp slik:</p> <ul style="list-style-type: none"> <li>i. Hovedprosjekt</li> <li>ii. Underprosjekt</li> <li>iii. Dokumenter</li> </ul> <p>Denne typen hierarki er tilpasset den konvensjonelle oppbygningen av både studioalbum og settlist (dette utdypes nærmere i seksjon 7.3.3).</p>
3. Synkron tekstkommunikasjon	Programmet skal tilby et direktemeldingsvindu (chat) der alle brukere skal kunne kommunisere med hverandre i sanntid. Her



		listes også alle som er innlogget i systemet. Alle skal selv kunne velge status (online, borte, opptatt) som vises bak navnet deres. Ved å høyreklikke på en bruker kan man få vist mer detaljert informasjon om brukeren.
4. Asynkron tekstkommunikasjon		<b>Alle prosjekter som er organisert under et hovedprosjekt skal kunne diskuteres/kommenteres av brukerne.</b> Denne diskusjonen blir logget og forblir synlig til alle andre brukere i samme gruppe.
5. Medieavspilling		<b>Alle musikkdokumenter skal kunne spilles av i programmet.</b> Brukeren skal kunne spole frem og tilbake i dokumentet, samt kontrollere lydstyrken på avspillingen. Mediespilleren lokaliseres i prosjektutforskeren for lett tilgang.
6. Versjonslogg		<b>Alle endringer på filer/dokumenter må dokumenteres av brukeren som forårsaket endringen.</b> Systemet viser et lite vindu (versjonslogg) når brukeren har endret et dokument, der brukeren må beskrive endringen han har utført. Her kan brukeren også få tilgang til tidligere dokumentasjon av endringer på det samme dokumentet.
7. System- og historielogg		<b>Viktige brukerhandlinger blir loggført automatisk av systemet og vises i et eget vindu.</b> Dette omfatter endring av dokumenter, opprettelse av nye dokumenter og sletting av dokumenter. Brukeren kan velge om han vil at denne oversikten skal være tilgjengelig hele tiden.
8. Ekstern redigering		<b>Programmet skal tilrettelegge for å kunne åpne og redigere dokumenter i tredjehåndsapplikasjoner.</b> Brukere som redigerer et dokument i tredjehåndsapplikasjoner blir vist under "also editing" i prosjektutforskeren hos de andre brukerne.
9. Synkronisering av brukernes musikkmateriale		<b>Alle brukernes dokumenter blir synkronisert med serveren.</b> Dette skjer automatisk ved oppstart og etter hvert som brukeren utforsker prosjektene som han tar del i.
10. Dokument-overvåkning		<b>Dokumenter blir overvåket av systemet. Når et dokument har blitt endret av brukeren vises dette under prosjektutforskeren, og brukeren må da velge "checkout/upload" for å bekrefte endringen.</b> Etter at endringen har blitt bekreftet, blir dokument(ene) lastet opp til serveren. Versjonsloggen må oppdateres av brukeren før

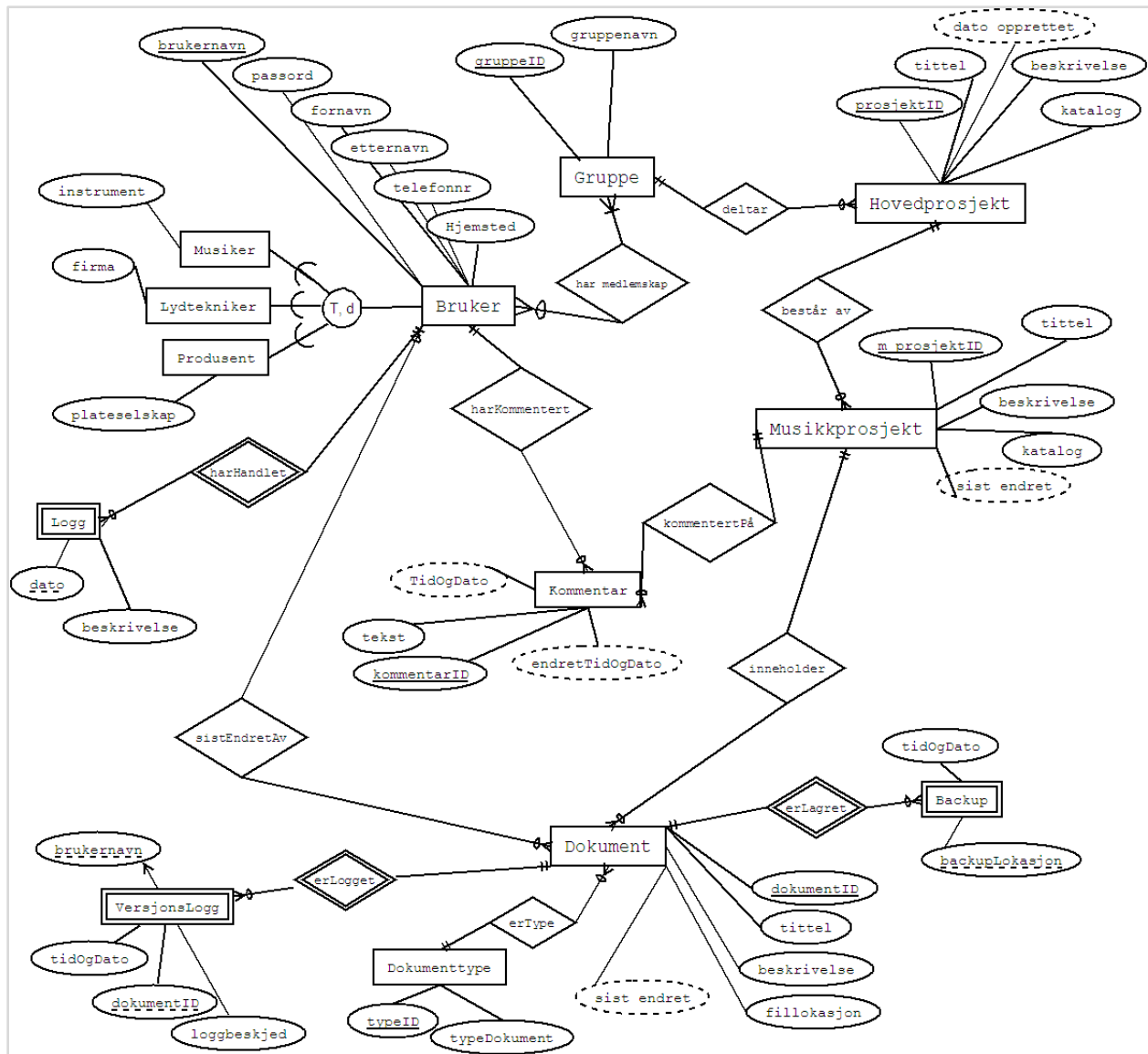
	opplastingen blir utført.
<b>11. Dokument-informasjon</b>	<b>Brukeren skal kunne vise og endre detaljer om hvert prosjekt/dokument.</b> Dette innebærer også visning av versjonsloggen til et gitt dokument.

### 7.2.2 Design

Designfasen hadde to primære holdepunkt. Det første dreide seg om å modellere det teknologiske fundamentet til applikasjonen, og det andre dreide seg om hvordan man kan kunne abstrahere funnene fra datainnsamlingen og forskningslitteraturen til å bli reflektert i applikasjonen for best mulig brukskvalitet. Det teknologiske fundamentet omfatter arkitekturen til applikasjonen, med andre ord hvordan man prosesserer data, hvordan applikasjonen sender data og hvordan man lagrer data. Lagringen av data er delvis gjennomført mot en database og delvis på et FTP-lager. Databasen ble først modellert ved bruk av EER-modellering<sup>37</sup>. Dette er vist i følgende figur:

---

<sup>37</sup> Enhanced Entity Relation. Se Elmasri & Navathe (2003) for en utdypende beskrivelse.



Figur 7 – EER-modell av databasen.

Jeg skal kort redegjøre for de viktigste aspektene ved modellen. Noen av aspektene ble indirekte introdusert i brukerkravene tidligere, men her vil sammenhengen mellom datalagring og funksjonalitet bli tydeligere. Ut fra modellen kan man se at en bruker kan være relatert til mange grupper og omvendt. Med *gruppe* menes to eller flere personer som inngår i et samarbeid. (Det betyr ikke nødvendigvis band, kvartett e.l., men kan også være en artist eller en gruppe uten navn. Poenget er at det eksisterer et samarbeid mellom aktørene som inngår i gruppen). Alle innenfor en gruppe får tilgang til hovedprosjektene, underprosjektene (*musikkprosjekt* i modellen) og dokumentene som er relatert til den gruppen. Prosjektene kan ikke deles mellom

flere grupper. (Dette er gjort for å minske antallet av feil som kan inntreffe). Men en bruker kan delta på flere grupper og derav få tilgang til musikkmaterialer på tvers av grupper. Hvert musikkprosjekt kan kommenteres av alle brukerne så mange ganger de ønsker. *Dokument* er den entiteten med flest relasjoner. Dokument-entiteten har blant annet to svake entiteter, henholdsvis *backup* og *versjonslogg*. Førstnevnte inneholder dokumentlokasjonen til alle tidligere versjoner som har blitt lagret av samme dokument. Sistnevnte, versjonslogg, inneholder tekstbeskrivelser av endringer og handlinger som er gjort mot et spesifikt dokument. Svake entiteter er relatert til maks 1 annen entitet. Dette er for å forhindre tap av dataintegritet i databasen. For eksempel vil en versjonslogg-tupple i databasen alltid være relatert til maks en dokument-entitet. Dokument-entiteten er i tillegg relatert til *dokumenttype*. Denne entiteten inneholder typen dokument som er lagret lokalt slik at systemet raskt finner ut hvilken funksjonalitet som skal tilbydes for et gitt dokument, eksempelvis medieavspilling, tekstfremvisning, etc. Dokumenttype innebærer med andre ord data som går igjen (for eksempel dokumenter av typen MP3), og for å forhindre dataredundans har dette blitt modellert i en egen entitet. *Logg* er en svak entitet relatert til brukeren. Her blir alle handlinger som har blitt forårsaket av brukeren lagt inn. Handlingen omfatter i hovedsak tillegging, endring eller sletting av data fra systemet. Hver logg knyttes opp til maks 1 bruker.

EER-modellen ble videre omgjort til et relasjonsskjema (se appendiks C for detaljer). Hver relasjon i skjemaet er normalisert på 3NF. Et relasjonsskjema er på 3NF *hvis* for alle avhengigheter på formen  $X \rightarrow [A]$  så gjelder minst en av følgende:

- X er en supernøkkel
- A er en hovedattributt
- $A \in X$

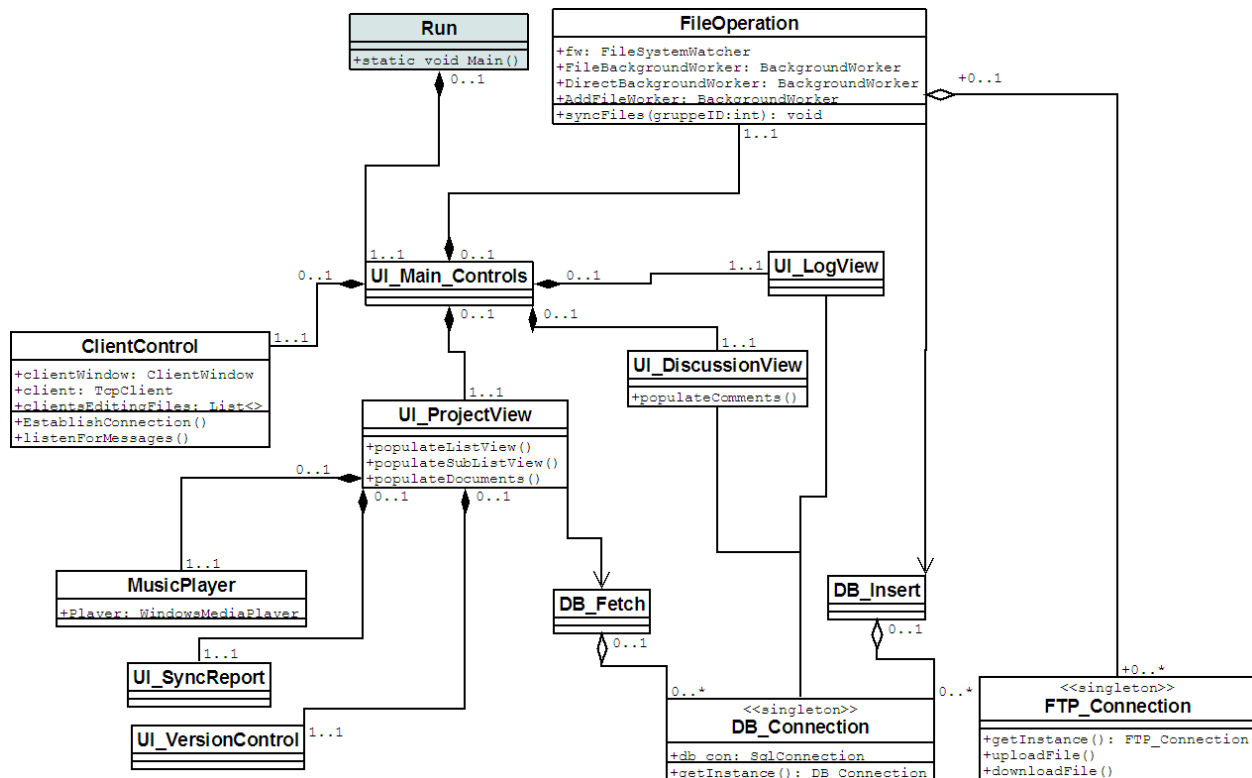
Poenget med normalisering er å forhindre dataredundans og uekte tupler. Dataredundans forekommer likevel (med hensikt<sup>38</sup>) i loggtabellen (se logg-entiteten i EER-modellen). Loggtabellen (Obs: ikke det samme som versjonslogg) logger hver handling som blir foretatt av en bruker. Her lagres tekstinformasjon om hvilken handling han har utført og på hvilken del av systemet den har blitt utført. Tabellen inneholder ingen referanser til prosjekter eller dokumenter, men kun tittel på det prosjektet eller dokumentet som er berørt av handlingen. Dette fordi at hvis

---

<sup>38</sup> Noen ganger omtalt som *trade off*.

en bruker sletter et dokument, både fildokument og referanse i databasen, så skal man kunne logge at dette har skjedd. (Slik at andre brukere får kunnskap om eksempelvis hvorfor en fil ikke finnes i systemet lengre). De andre brukerne kunne ikke fått kunnskap om hvilke filer som hadde blitt slettet dersom man hadde en peker til en tuppel som hadde blitt slettet fra databasen. Referanser til data i en ikke-eksisterende tabell er ugyldige og poengløse.

Objektorientert modellering av applikasjonens oppbygning ble utført ved bruk av UML<sup>39</sup>. Det er ikke hensiktsmessig å dedikere for mye tid på utgreiing om arkitekturen bak applikasjonen. Dette fordi det totale antallet av klasser, variabler og metoder er relativt høyt<sup>40</sup>. I stedet har det blitt konstruert en abstrahert UML-modell over systemarkitekturen, der kun et utvalg av de mest essensielle klassene, attributtene og metodene har blitt inkludert.



Figur 8 – Abstrahert UML-modell som viser systemarkitekturen.

UI\_Main\_Controls er en kontrollklasse som sammenbinder resten av GUI-klassene og samkjører operasjoner mellom dem. Alle klassene som har UI som prefiks innebærer at de er delvis en

<sup>39</sup> Unified Modeling Language

<sup>40</sup> Antall linjer med programmeringskode har overstegit 14'000

brukergrensesnittklasse og delvis en funksjonalitetsklasse<sup>41</sup>. De som ikke har UI som prefiks er kun funksjonalitetsklasser. DB\_Connection-klassen oppretter en tilkobling til SQL Server 2005 Express-databasen. Denne er designet etter singleton-mønsteret som innebærer at man kun kan opprette en tilkobling til databasen. Dette betyr i praksis at alle klassene som bruker DB\_Connection får samme instans av det opprettede objektet. Dette gjelder også FTP\_Connection-klassen. Denne oppretter en FTP-tilkobling til serveren for opplasting og nedlasting av dokumenter. All opp- og nedlasting foregår på egne tråder (det vil si at ned- og opplastingsmetodene arbeider i bakgrunnen) for ikke å stagnere systemet mens synkronisering pågår. FileOperation-klassen inneholder også en metode som kontinuerlig sjekker om brukeren har endret på dokumentene som er relatert til hans gruppe.

### 7.2.3 Koding og implementering

Programmeringen ble gjennomført i programmet Visual Studio 2005. Her ble det konstruert to bibliotek og to applikasjoner. Bibliotekene omfatter et database- og et direktemeldingsbibliotek. Applikasjonene omfatter CMO (som blir installert på klienten) og et direktemeldingsprogram som håndterer innkommende direktemeldinger på serveren. Direktemeldingsprogrammet delegerer meldinger ut til rette vedkommende. Direktemeldingene blir streamet over en TCP/IP-protokoll fra klientene til serveren, og videre sendt ut fra serveren til klienter som er medlem av samme gruppe. I tillegg håndterer programmet en oversikt over hvilke handlinger de forskjellige samarbeidsdeltakere utfører i CMO, for eksempel hvilken fil som blir redigert, samt hvem som redigerer den.

Testing ble gjort underveis for å forsikre stabilitet i applikasjonen. Noen initiale stabilitetstester ble utført med andre masterstudenter før den egentlige evalueringen tok sted. Se appendiks A for informasjon om tilgang til kildekoden.

### 7.2.4 Evaluering

Groupware Walkthrough (GWW) ble gjennomført, som nevnt, sammen med to brukere. Den ene var aktiv gitarist, komponist og produsent. Den andre var aktiv vokalist med erfaring fra både

---

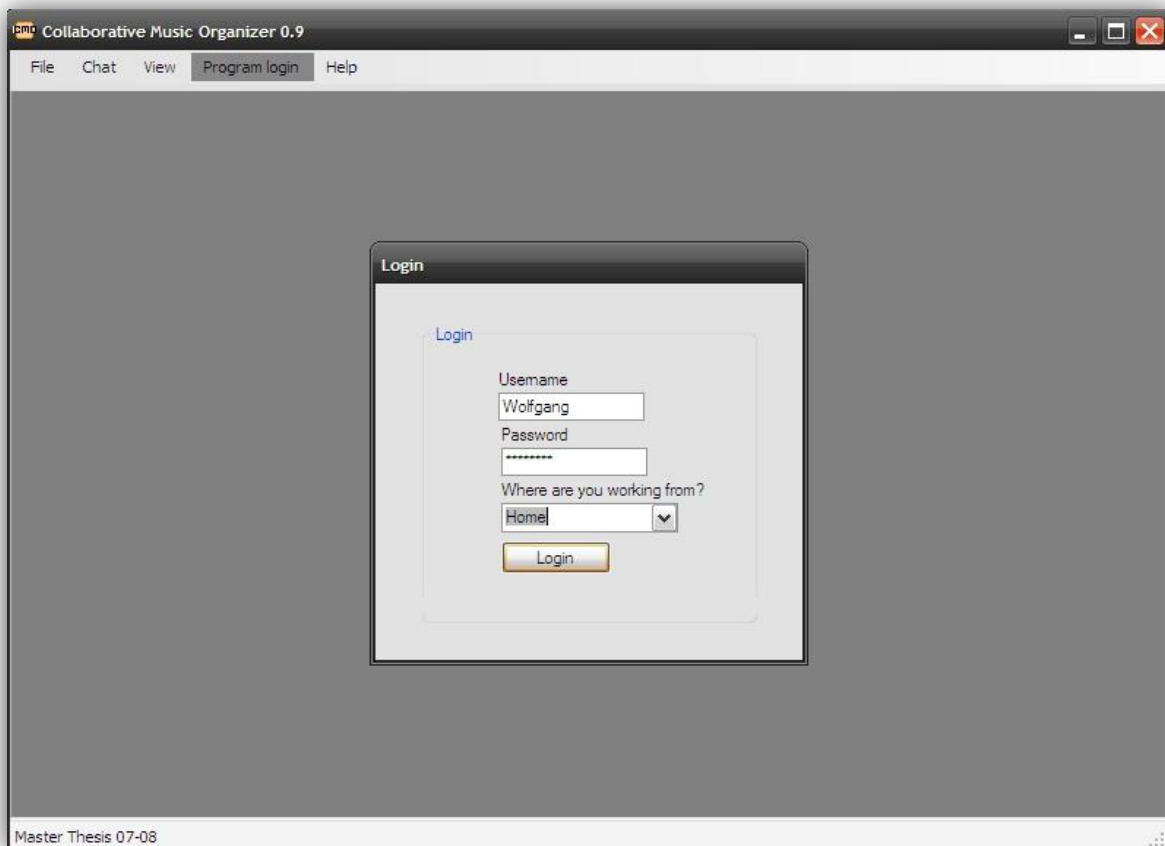
<sup>41</sup> C#-språket benytter partielle klasser for å forenkle sammenkoblingen av brukergrensesnitt- og funksjonalitetsklasser. Dette innebærer at begge klassene benytter samme klassenavn, og at hver klasse blir definert partiell, men er separert i forskjellige filer.

kor og flere band. Det ble gjort to gjennomganger av de samme oppgavene, en med hver bruker. Jeg satt i samme rom som brukerne, men ved hver vår PC. Brukerne hadde på forhånd fått en liten innføring i basisfunksjonaliteten til CMO. Gjennom evalueringen var undertegnede evalueringsansvarlig og testbruker. Sammen prøvde vi å utføre et sett av samarbeidsoppgaver i applikasjonen. Oppgavene innbefattet både individuelle oppgaver og samarbeidsoppgaver. Jeg var altså deltaker og observant under evalueringen, og svarte på spørsmål dersom brukeren lurte på noe. Alle spørsmål ble innsamlet slik at de kunne vurderes i ettertid. Når evalueringen var ferdig, gjenstod det å identifisere hvorvidt oppgavene ble løst med suksess. Spørsmålene om brukertilfredshet var relatert til hver oppgave som skulle løses, og målet her var å finne ut om det var grunn til å være tilfreds med prosessene som ble utført på hver oppgave, samt utfallet av handlingene på hver oppgave. Funnene fra evalueringen blir gjennomgått i kapittel 8.

### **7.3 Designvalg: Gjennomgang av Collaborative Music Organizer**

I denne seksjonen skal jeg gå igjennom designvalgene som har blitt foretatt under konstruksjonen av CMO. Valgene er basert på teorien som ble presentert i kapittel 3, samt funn fra datainnsamlingen. Det skal blant annet vises hvordan awareness er ivaretatt gjennom designet av applikasjonen, hvordan viktige aspekter ved CIS er gjenspeilet i applikasjonen, samt hvordan samarbeidsmekanismene er underbygget. Å underbygge samarbeidsmekanismene er, som nevnt tidligere, helt avgjørende for at basisaktivitetene i et shared workspace er gjennomførbare, og det er som nevnt samarbeidsmekanismene som utgjør grunnlaget for evalueringen. Under presentasjonen av CMO blir det benyttet tre eksempelbrukere: Wolfgang, Bjarte og Bach.

### 7.3.1 Innlogging



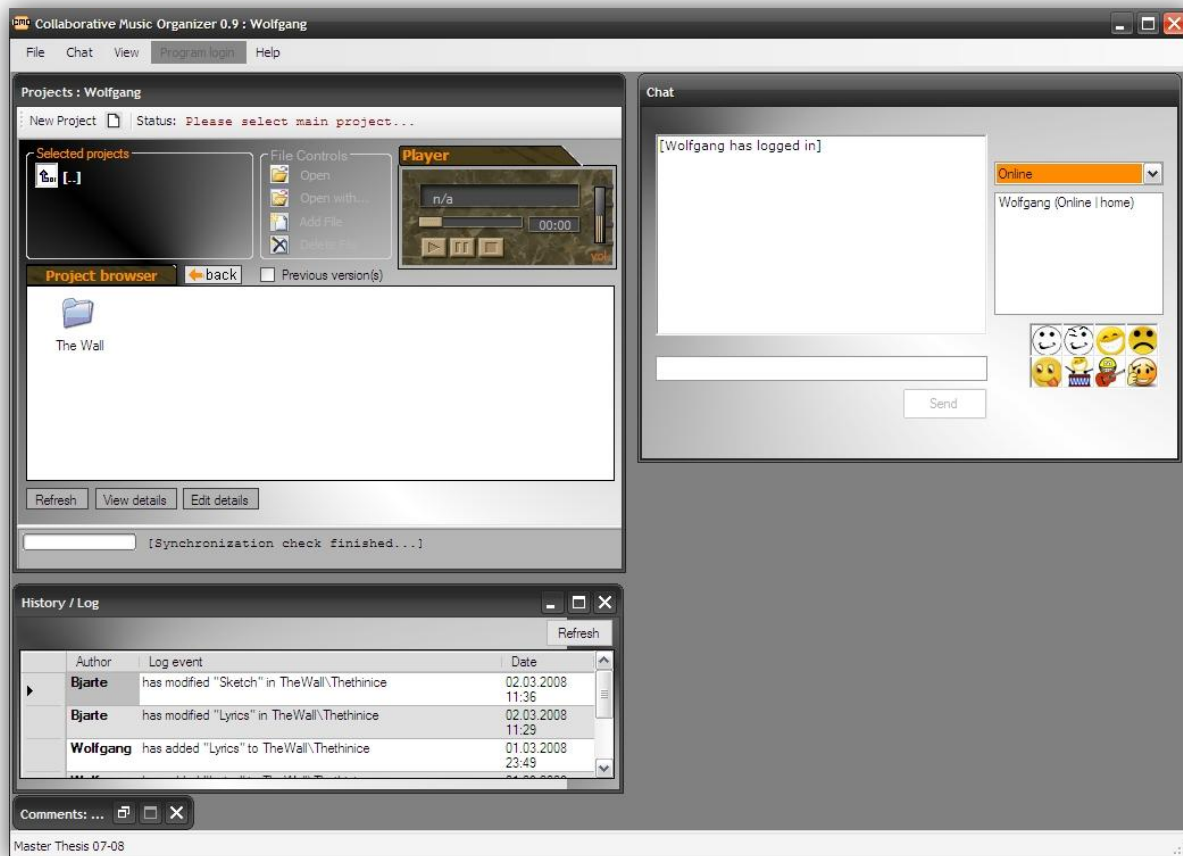
Figur 9 – Skjerm bilde av innloggingsvinduet.

Dette er det første bildet som blir presentert til brukerne når de starter applikasjonen. Her må alle brukerne skrive inn sitt brukernavn, passord og hvor de arbeider fra. Sistnevnte felt har blitt valgt for å stimulere brukernes awareness om hvor sine samarbeidspartnere befinner seg. Her kan man eksempelvis skrive inn hjemme, studio, fra utlandet, toget, og så videre. Informasjonen om hvor brukerne befinner seg blir presentert bak navnet til alle som er innlogget i systemet. Listen over alle som er innlogget er synlig for brukerne umiddelbart når de har logget inn. Grunnen til å velge en strikt innloggingsprosedyre er relatert til beskyttelse av brukernes arbeid. Det antas at brukerne ikke ønsker å ha arbeidet sitt offentlig tilgjengelig. Dette er også relatert til en av samarbeidsmekanismene til Gutwin & Greenberg (2000). Mekanismen kalles *beskyttelse* og går ut på at brukerne av et gitt system skal være beskyttet mot ødeleggelse av arbeidet sitt. Ved å ha musikk materialet tilgjengelig for alle brukerne i systemet, på tvers av ulike grupper, kunne hærverk eller uforsiktighet av utenforstående brukere medført at arbeid gikk tapt. Krav om



brukerautentifisering forhindrer imidlertid dette problemet i å inntreffe. Innloggingsprosedyren blir også benyttet for å bestemme gruppetilhørigheten til hver bruker.

### 7.3.2 Oversiktsbilde



Figur 10 – Skjermbilde som viser hele programmet etter innlogging.

Når man har logget inn i applikasjonen med et gyldig brukernavn og passord kommer man til bildet som er vist i figur 10. Vinduet øverst til venstre viser prosjektutforskeren, og vinduet til høyre viser chat-vinduet. I førstnevnte vindu får man mulighet til å utforske, etablere, endre, spille av, vise detaljer og se versjonslogg av prosjektene og dokumentene som tilhører gruppen man er medlem av. Chat-vinduet utgjør grensesnittet der brukerne kan interagere synkront. Her vises også alle brukerne som er innlogget, deres status og hvor de befinner seg. Smileikoner<sup>42</sup> har blitt inkludert siden dette er et utbredt konsept innenfor direkte meldingsfunksjonalitet. Nederst til venstre i applikasjonsoversikten ser man loggvinduet og et minimert kommentarvindu.

<sup>42</sup> Eng.: Smiley

Kommentarvinduet brukes til asynkron kommunikasjon om prosjektene (for eksempel diskusjoner og planlegging), og loggvinduet gjenspeiler de viktigste handlingene som har blitt utført i systemet av alle brukerne. På toppen av applikasjonsvinduet finner man menyvalg som File, Chat, View, Program Login og Help. Gjennom disse menyvalgene kan man avslutte programmet, endre på oppkoblingsdetaljer (IP, Port og brukernavn), tilpasse vinduer og vise hjelp. Ved å klikke hjelp blir det presentert et vindu som forklarer basisfunksjonaliteten ved programmet. Jeg skal gå nærmere inn på detaljene om de forskjellige vinduene etter hvert.

### 7.3.3 Prosjektutforskeren



Figur 11 – Skjerm bilde av prosjektutforskeren.

I figur 11 ser man prosjektutforskeren sett fra brukeren som i dette eksemplet kalles Wolfgang. Under «Selected Projects» (øverst til venstre) vises hierarkiet man befinner seg i. Akkurat nå har Wolfgang åpnet hovedprosjektet som heter «The Wall», og underprosjektet som heter «The thin ice». I «Project browser» (midten) ser man at det ligger fire dokumenter her. Denne hierarkiske

oppbygningen har blitt valgt på bakgrunn av eksisterende omstendigheter innenfor eksisterende musikkpraksis. Hovedprosjekter kan for eksempel utgjøre et studioalbum. På hvert album er det flere låter. Disse kalles underprosjekter i CMO. På hvert underprosjekt finnes musikkmateriale. Dette utgjør, som nevnt tidligere, musikkfiler, tekster og noter. Hovedprosjektet trenger ikke nødvendigvis være et studioalbum. Et annet eksempel på hovedprosjekt er en konsert. På hver konsert blir det fremført flere låter, det vil si underprosjekter i CMO, og på hvert underprosjekt ligger tekstene og musikken som skal fremføres. Det er tenkt at denne kjente hierarkiske oppbygningen er lett å gjøre seg kjent med for brukerne.

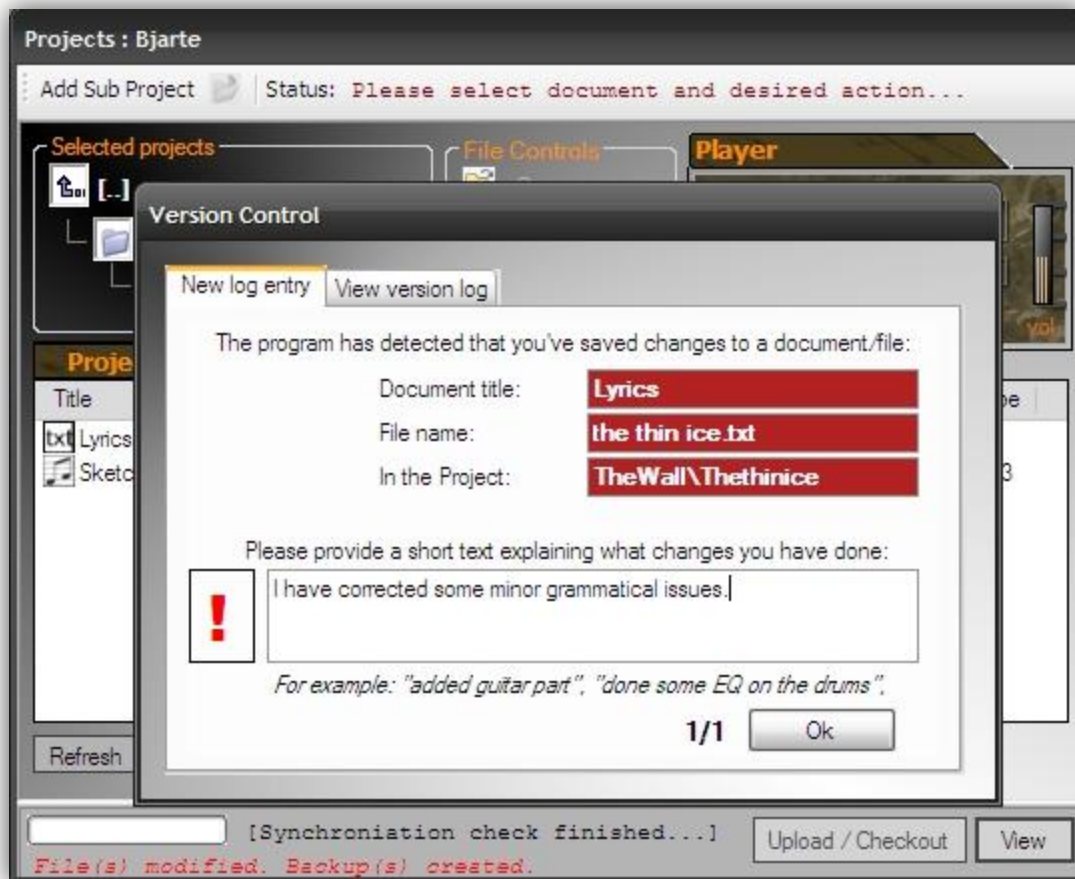
På prosjektuforskerbildet ser man at det finnes noen dokumenter med rød skrift og gjennomgående strek. Dette symboliserer at dokumentene er tidligere versjoner. Man kan velge selv om man ønsker å vise tidligere versjoner ved å merke *previous version(s)*-feltet. «File Controls» (øverst i midten) illustrerer handlingene man kan utføre på dokumentene. Hvis man klikker *add file* presenteres en dialog der man spesifiserer tittel og beskrivelse på dokumentet man ønsker å legge til, og etterpå må man velge dokumentet man ønsker å legge til. Workspace-awareness (WA) har vært et betydningsfullt konsept under designet av prosjektutforskeren. I «Project browser» kan man se at opphavet til hvert dokument illustreres gjennom *by user*-feltet. Av denne informasjonen kan man resonnerer seg frem til at for eksempel lyrics-dokumentet ble først laget av Wolfgang, og så endret av Bjarte. Dette fordi lyrics-dokumentet, som har rød skrift, symboliserer at det er en tidligere versjon, og i *by user*-feltet på dokumentet med rød skrift står det Wolfgang.

Innenfor WA er det også sentralt å vise hva brukerne foretar seg på nåværende tidspunkt i applikasjonen. I feltet *also editing* kan man se at Bjarte redigerer på dokumentet «Lyrics». Hvis Wolfgang har tenkt å endre på dokumentet, så kan han først rådføre seg med Bjarte (gjennom chat-vinduet) for å forsikre seg om at de ikke gjør samme handlinger på samme dokument. Eller kanskje Wolfgang bare er interessert i å finne ut hva Bjarte gjør. *Also editing*-feltet underbygger samarbeidsmekanismen *koordinasjon av handling*. Informasjonen om hvem som redigerer et gitt dokument kan hjelpe brukerne å forhindre at de dupliserer handlingene sine. I tillegg kan *also editing* karakteriseres som samarbeidsmekanismen *indirekte kommunikasjon* siden dette er informasjon som blir gitt om en bruker uten at han gir den fra seg eksplisitt. Det er med andre

ord systemet som identifiserer data om brukeren, og presenterer dette som informasjon til de andre brukerne.

Hvis Wolfgang ønsker å se flere detaljer om et gitt dokument, kan han klikke på *view details*. Dette åpner et lite vindu som viser tittelen og beskrivelsen av dokumentet. Beskrivelsen blir lagt inn av den som først legger til dokumentet. Hensikten er å beskrive intensjonen med dokumentet, eksempelvis om det er et introparti til en komposisjon eller et uferdig tekstutkast. Denne informasjonen kan også endres i ettertid av hver bruker.

Når brukeren spiller av et mediedokument vil ikke informasjonen om brukerens avspilling presenteres til de andre brukerne. Det anses som unødvendig å vise alle brukerhandlinger av de andre deltakerne. Medieavspilling har ingen implikasjoner for brukerne i forhold til arbeidet deres, og hvis systemet ble designet slik at det gav informasjon om hver eneste brukeraktivitet, så kunne dette medført at viktig awareness-informasjon ikke ble oppfattet av brukerne. Det ville fort blitt mye informasjon å forholde seg til, og dette kunne også gjort det problematisk for brukerne å skille informasjon som var betydningsfull for koordinasjon av deres handlinger med informasjon som ikke var betydningsfull.

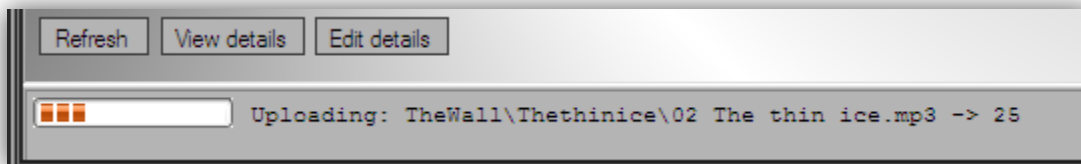


Figur 12 – Skjermbilde av versjonskontrollen.

Etter at en bruker har gjennomført en endring på et dokument kommer det opp en *upload /checkout*-knapp (se nederst på bildet, figur 12) med informasjon om at noen filer har blitt modifisert. I bildet ovenfor har brukeren foretatt en endring på dokumentet «Lyrics». Når brukeren klikket *upload/checkout* kom *Version Control* opp. *Version Control* bidrar til at alle endringer blir dokumentert slik at deltakerne kan orientere seg om hva som har skjedd i sitt shared workspace. Her bes brukeren om å gi en liten tekstlig beskrivelse av endringen han har gjort. Dette er spesielt viktig innenfor WA. Gjennom tekstlige beskrivelser av endringene får brukerne informasjon om hvilken handling som har blitt utført på et gitt dokument. Uten tekstlige beskrivelser kan brukerne observere at det har blitt utført en endring, men de får ingen informasjon om hva denne endringen innebærer. Dette medfører at brukerne må lete etter endringen og muligens sammenligne det endrete dokumentet med tidligere versjoner. Endringsbeskrivelser er av den grunn obligatorisk fordi det forenkler samarbeidet til deltakerne.

Brukerne kan også velge å se tidligere endringsbeskrivelser i *view version log* (tabulator ved siden av *New log entry*). Når brukerne må avgi informasjon om sine handlinger, underbygger dette samarbeidsmekanismene *planlegging* og *overvåkning*. Denne informasjonen bistår brukerne i planlegging av sine aktiviteter basert på hva som eksempelvis har blitt gjennomført på tidligere versjoner av dokumentene. Overvåkning betyr at brukerne hele tiden har mulighet til å være oppdatert på informasjon og handlinger som har blitt utført av sine meddeltakere.

Hvis brukeren har endret flere dokumenter før han valgte å klikke upload/checkout, så illustreres dette i under antallet dokumenter han har endret. Versjonskontrollen krever tekstlige beskrivelser om hvert dokument som har blitt endret. Dersom brukeren ønsker å se hvilke dokumenter han endret, kan han klikke på *View*-knappen (nederst til høyre i figur 12).



Figur 13 – Utsnitt av prosjektutforskeren.

I figur 13 ser man et utsnitt av prosjektutforskeren. Her vises opplastingsprosessen etter at en bruker har klikket upload/checkout. Dette gir informasjon om at filen er i ferd med å bli distribuert til nettverkets sentrale server. Filene har ikke nødvendigvis samme navn som tittelen på dokumentet. Tittelen blir bestemt av brukeren som oppretter dokumentet.

I følge brukerkravene skal brukerne bli synkronisert med serveren automatisk hver gang applikasjonen startes. Denne funksjonaliteten er implementert, og synkroniseringsinformasjonen blir presentert samme sted som opplastingsprosessen. Synkronisering skjer også forløpende når en bruker utforsker forskjellige prosjekter. Når opplasting/synkronisering er ferdig, vises det en beskjed om dette. I prosjektutforskeren, som vist tidligere, så man at det stod «Synchronization check complete». Dette viser at systemet har gjennomført en synkronisering med serveren, og at alle felles prosjekter og dokumenter har blitt synkronisert.

### 7.3.4 Chat-vinduet



Figur 14 – Skjermbilde av chat-vinduet.

Ovenfor ser man et bilde av chat-vinduet. Dette utgjør hovedgrunnlaget for synkron kommunikasjon i applikasjonen. Av interaksjonen kan man lese at Wolfgang har identifisert en endring som ble utført på et dokument han hadde opprettet. Denne informasjonen fikk han først fra loggen til systemet, etterfulgt av ytterligere detaljer gjennom versjonsloggen til dokumentet. Chat-vinduet legger opp til at brukerne kan diskutere intensjonene med handlingene sine og på den måten øke sin forståelse av sine felles informasjonsobjekter. Innenfor CIS er det sentralt at aktørene kan diskutere meningen av informasjonsobjektene i arbeidsområdet. Dette bidrar til å øke forståelsen av konteksten som informasjonsobjektet kom fra, og hvordan forskjellige brukere tolker informasjonsobjektet. Når man har mulighet til å diskutere forskjellige tolkninger, kan man komme nærmere en felles tolkning, eller delt enighet som Schmidt og Bannon (1992) nevner (se seksjon 3.4).

Chat-vinduet underbygger flere av samarbeidsmekanismene til Gutwin & Greenberg. Direktemeldingsfunksjonalitet tilrettelegger for at samarbeidsdeltakerne kan kommunisere ved bruk av *eksplisitt kommunikasjon*. Funnene fra intervjuene viste også at flere av komponistene

allerede var kjent med direktemeldinger og benyttet slike tjenester for å kommunisere med sine samarbeidspartnere. Å bruke det som allerede er kjent for de potensielle brukerne av systemet kan bidra til at både opplæring og terskelen for å anvende funksjonaliteten er lavere. *Hjelp* er en annen viktig samarbeidsmekanisme. Dette innebærer at deltakerne kan be om hjelp og tilby hjelp til sine meddeltakere. Gutwin & Greenberg (2000) nevner at eksplisitt kommunikasjon kan utgjøre grunnlaget for å underbygge denne samarbeidsmekanismen. Dette har også vært en av intensjonene med chat-vinduet.

Innenfor WA er det viktig for en bruker å vite hvem som er aktiv i et shared workspace. Til høyre på bildet av chat-vinduet blir dette underbygget ved å vise en liste over de brukerne som er innlogget. Hver bruker har mulighet til å velge sin egen status. De kan eksempelvis markere om de er tilgjengelig eller opptatt, og denne informasjonen kan påvirke hvordan de andre forholder seg til dem. Hvis statusen til Wolfgang er satt til *away* får andre brukere informasjon om at Wolfgang ikke er tilstede akkurat nå. Det innebærer at dersom de henvender seg til Wolfgang, så er det mulig de ikke får svar. Og dette er ikke nødvendigvis fordi Wolfgang ikke vil snakke med dem. Når en bruker logger inn eller ut blir dette også vist som informasjon i chat-vinduet. For eksempel: «Wolfgang has logged out». I tillegg ser man at bak statusen til brukeren finnes det ytterligere et substantiv. Dette substantivet uttrykker hvor brukeren befant seg når han logget inn. På bildet kan man se at bak Wolfgang står det «Home». Dette indikerer at han jobber hjemmefra. Denne informasjonen ble, som nevnt tidligere, lagt inn av brukeren på innloggingssekvensen. Det antas at hver bruker vil forstå hva denne informasjonen betyr. Wolfgang kan se at det han skrev inn på innloggingssiden under spørsmålet «Where are you working from?» står bak hans navn og status. Da kan man anta at han forstår hva dette indikerer bak sine meddeltakere.



The screenshot shows a window titled 'User information'. Inside, the role 'Musician' is displayed in bold, with 'Instrument: Piano' below it. There are four input fields: 'Name' with the value 'Wolfgang', 'Surname' with 'Mozart', 'Phone' with '99443322', and 'Home' with 'Salzburg'.

Figur 15 – Skjerm bilde av bruker-informasjonsvinduet.

Awareness-konseptet påpeker at det er viktig å gi informasjon om menneskene som eksisterer rundt en. I CMO er ikke nødvendigvis brukernavnet alene tilfredsstillende for å kunne forstå hvem og hvor brukerne er fra. Som et supplement kan brukerne høyreklikke på hverandre og skaffe seg ytterligere informasjon. I figur 15 ser man et eksempel på dette. Her vises navn, etternavn, telefonnummer og hjemsted til Wolfgang. Et av intervjuobjektene sa at han hadde erfaring fra samarbeid med internasjonale musikere.

Informasjon om hjemsted er kanskje av mer betydning i disse situasjonene enn for musikere der alle kommer fra Bergen. Det kan også tenkes at alle som samarbeider på et gitt prosjekt ikke nødvendigvis er bekjente. For eksempel hvis et band leier inn en frilanser for å spille et instrument som kun brukes på en komposisjon. I slike scenarioer kan denne funksjonaliteten være nyttig.

Øverst i figur 15 kan man se at rollen til brukeren blir angitt med store bokstaver. Rollen indikerer hvorvidt personen er musiker, lydtekniker eller produsent. I tillegg blir det uttrykt hvilket instrument han spiller eller hvilket firma/selskap han er fra. Det ble nevnt i seksjon 3.4 at forskjellige mennesker med ulik bakgrunn og kompetanse kan problematisere hvordan de oppnår en felles forståelse for informasjonsobjektene i et CIS. Det er tenkt at denne tilleggsinformasjonen (om rollen til hver aktør) kan påvirke hvordan brukerne forholder seg til hverandre. Kanskje vil en lydtekniker forenkle tekniske uttrykk hvis han vet at han henvender seg til en klassisk fiolinist. Vinduet med brukerinformasjon er i den hensikt et forsøk på å forbedre forholdene for brukernes bevissthet om sine meddeltakere. Det innebærer altså at de lett skal kunne orientere seg om hvilken bakgrunn og kompetanse meddeltakerne har.

Telefonnummer er, som man ser, en del av brukerinformasjonen som er tilgjengelig for andre brukere. Hovedgrunnen til å inkludere telefonnummer er at flere av intervjuobjektene beskrev telefon eller muntlig kommunikasjon som deres hovedpreferanse for kommunikasjon. Ved å inkludere telefonnummeret til brukerne, viser man at det ikke er ønskelig å tvinge ny teknologi

på brukerne, men at de også kan bruke sine eksisterende kommunikasjonsmedier som supplement til denne applikasjonen. Det kunne tenkes at to brukere, som hadde vanskelig for å kommunisere sine meninger gjennom tekstbaserte direktemeldinger, ville foretrukket en telefonsamtale. Denne problemstillingen blir også omtalt av Schmidt og Bannon (se seksjon 3.1). Her nevner de nettopp viktigheten av å ikke forstyrre eller forhindre aktivitetene slik de tradisjonelt fungerer. Dette er altså intensjonen. Gjennom applikasjonen får man tilgang til ny kommunikasjonsfunksjonalitet, men samtidig neglisjerer man ikke at mange benytter telefonen som kommunikasjonsmedium i eksisterende praksis. I tillegg burde man ha lagt inn et felt som inneholdt e-postadressen til brukerne, siden e-post var et mye brukt kommunikasjonsmedium i eksisterende praksis.

### 7.3.5 Kommentarfvinduet



Figur 16 – Skjerm bilde av kommentarfvinduet.

Kommentarfvinduet sin hovedfunksjon er å tilby asynkron kommunikasjon. Vinduet har fått dette navnet for å skape en mest mulig vid forståelse av hva det kan brukes til. Ved å bruke begrepet *kommentarer* forsøker man ikke å begrense samarbeidsdeltakernes bruksområder av vinduet.

Gjennom dette vinduet kan de for eksempel planlegge mål, fordele arbeid, koordinere oppgaver og diskutere musikkmateriale. Hvert underprosjekt har muligheter for å kommenteres. Asynkron kommunikasjon er mest hensiktsmessig når de som inngår i samarbeidet ikke er innlogget til

samme tid. Tidligere ble det nevnt at det i enkelte sjangre var unødvendig at alle deltakerne i en samarbeidsgruppe deltok i komposisjonssamarbeidet samtidig, men at noen av deltakerne kunne møtes for å komponere musikkmateriale sammen (se seksjon 2.2). Hvis disse deltakerne benytter kommentarvinduet til å legge inn plan for videre progresjon, kan kommentarvinduet bidra til å oppdatere de som ikke deltok på samarbeidet om hva som bør gjøres videre.

I figur 16 ser man hvordan brukeren Bach har sagt at han jobber på en introduksjon til prosjektet «The thin ice», og at Wolfgang har antydnet at han skal se på refrenget. Kommentarfvinduet underbygger samarbeidsmekanismene *koordinasjon av handling og planlegging*. Koordinasjon og planlegging er også viktige momenter i articulation work-konseptet. Opphavet til hver kommentar er tydelig skildret ved brukernavn og tidspunkt for kommentaren. De nyeste kommentarene er sortert øverst, men dette kan man overstyre hvis annen sortering er ønskelig. Man har lett tilgang til å publisere nye kommentarer ved å klikke på *new entry*.

### 7.3.6 Loggvinduet



Figur 17 – Skjerm bilde av loggvinduet.

Loggvinduet (se figur 17) viser de viktigste handlingene som blir utført i systemet. Her blir opphavet til handlingen, tidspunktet for handlingen og en tekstlig beskrivelse av handlingen registrert. Dette gir viktig awareness-informasjon om alt som forløper seg i det delte arbeidsområdet. Alle brukerne får på denne måten kjapp tilgang til de viktigste handlingene som har blitt utført av andre brukere, uten at de trenger å gå inn på hvert prosjekt for å sjekke om dokumenter eksempelvis har gjennomgått endringer. Man kan si at dette støtter

samarbeidsmekanismen *indirekte kommunikasjon* og *overvåkning*. Brukerne etterlater seg informasjon indirekte som bidrar til at de kan overvåke hverandres handlinger. Informasjonen genereres med andre ord automatisk av systemet dersom brukerens handling har implikasjoner for de andre brukernes arbeid. Men hva betyr det at noe har implikasjoner for de andre brukernes arbeid? I denne sammenhengen omfatter dette kun opprettelse, endringer og sletting av musikkmateriale. Man kunne hevde at en kommentar har implikasjoner for de andres arbeid hvis denne kommentaren bidro til å klargjøre et eller annet aspekt som var avgjørende for å skape en felles forståelse. Men ulempen er at dersom kommentarer skulle vært del av loggen ville dette bidratt til å skape en mer uoversiktlig logg. Det kunne fort føre til at opprettelse eller endringer av musikkprosjekter ble oversett. Samtidig kan en bruker, som observerer en endring eller opprettelse av musikkmateriale i loggen, umiddelbart navigere til dette prosjektet og kjapt orientere seg om det har blitt lagt til noen kommentarinlegg. (Kommentarvinduet kommer automatisk frem når man går inn på et underprosjekt). Samlet utgjør dette grunnlaget for å ekskludere kommentarinlegg fra loggen. Dersom man ønsket å logge nye kommentarinlegg, kunne dette blitt organisert under en egen del av loggen. Og på den måten ville endringer, opprettelse og sletting av musikkmateriale blitt separert fra informasjonen om nye kommentarer.

### **7.3.7 Tilleggsmomenter om designvalgene**

Her skal det kort redegjøres for brukerkrav som ikke har blitt implementert, samt hvorfor de ikke har blitt implementert. Grunnene til at noe funksjonalitet har blitt utelatt skyldes teknologiske vanskeligheter og tidsaspektet for studiet. I prosjektutforskeren så man at det fantes en knapp for sletting av dokumenter. Denne funksjonen har ikke blitt implementert på grunn av at dette er en relativt omfattende operasjon, og funksjonen ble nedprioritert i forhold til annen funksjonalitet (som har blitt gjennomgått i tidligere seksjoner). Man kunne raskt laget en funksjon som slettet referansene fra databasen og slettet filen fysisk fra klientene og serveren. Men når mange brukere samarbeider om felles arbeid kan dette fort by på problemer. Hva skjer hvis en bruker sletter et helt prosjekt som har krevd hundrevis av felles arbeidstimer? Man kan tenke seg til at en slik handling ikke blir varmt mottatt fra de andre brukerne. Hvordan kunne man implementert denne funksjonaliteten på en hensiktsmessig måte?

Samarbeidsmekanismen *beskyttelse* ble introdusert tidligere, og går ut på å beskytte deltakerne fra at andre deltakere sletter eller ødelegger deres arbeid. Systemet må gå aktivt inn for å forhindre at slike tilfeller forekommer. Sletting av prosjekter og dokumenter faller inn under beskyttelsesmekanismen. En potensiell løsning på slettingsproblemet kunne vært å la hver bruker få rettigheter til å slette dokumenter som han har opprettet. Videre måtte andre brukere fått bekreftelse av denne opprettelsesbrukeren dersom de ønsket å slette noe som han hadde opprettet. Hvis en bruker derimot ønsket å slette et helt prosjekt (med alt musikk materialet som var relatert til dette), kunne systemet midlertidig lagt dette prosjektet et annet sted i det delte arbeidsområdet. Dersom prosjektet skulle slettes permanent kunne man kreve en konsensus av alle brukerne, det vil si at alle deltakerne på en gitt gruppe ble bedt om å bekrefte slettingsoperasjonen. Videre kunne hver bruker gjenopprette prosjektet hvis de ønsket det.

Beskyttelsesmekanismen har altså ikke blitt implementert når det gjelder sletting av andre arbeid, men problemet har ikke blitt neglisjert heller; det er tross alt ikke mulig for brukerne å slette arbeid fra systemet.

Et annet moment, som ble nevnt i brukerkravene, var muligheten til å endre komposisjoner ved hjelp av tredjehåndsapplikasjoner. Muligheten har blitt tilfredsstilt i noen grad, men fungerer ikke med store studiprogrammer som for eksempel Digidesign Pro Tools<sup>43</sup> og Cubase Studio<sup>44</sup> på nåværende tidspunkt. Disse programmene ble benyttet av noen av intervjuobjektene.

Grunnen til at disse programmene ikke er støttet, er at de ikke forholder seg til en etablert standard innenfor lydredigering. Programmene bruker derimot sine egne filformater som ofte består av en type prosjektfil som igjen peker til aktuelle wave- eller midifiler. Hvis man legger denne typen filer inn i CMO får man altså ikke med wave- og midifilene. Man kan likevel bruke for eksempel Cubase til å eksportere wave-filene til disk, for så å legge dem inn som wave-dokumenter i CMO, men da må man importere dem igjen i Cubase for videre utvikling utenfor CMO. En potensiell løsning på dette problemet er å finne ut hvordan disse prosjektfilene er oppbygd for så å lage en filleser som kan snappe opp hvor referansene til wave- og midifilene ligger. Deretter kunne man automatisert innleggelsen av disse filene i systemet når en bruker la

---

<sup>43</sup> Digidesign Pro Tools: <http://www.digidesign.com/index.cfm?navid=28&langid=151&>

<sup>44</sup> Cubase Studio 4: <http://www.steinberg.net/1008+M52087573ab0.html>

inn en prosjektfil. Dette kan likevel vise seg å være et krevende tiltak siden det ikke finnes noen garanti for at programvareleverandørene benytter samme oppbygning på prosjektfilene. Programmer som Wavelab, SoundForge, Notepad, Microsoft Word, og Sibelius er imidlertid støttet av applikasjonen, og disse gir tilstrekkelige redigeringsmuligheter for å kunne evaluere applikasjonen i forhold til redigering gjennom tredjehåndsapplikasjoner.

## **7.4 Klassifisering av Collaborative Music Organizer**

I seksjon 3.3 ble Weinbergs (2002) klassifiseringskategorier gjennomgått. Applikasjonen, som er utviklet i dette studiet, kan (basert på Weinbergs kategorier) klassifiseres som en kombinasjon av (4) The Construction Kit Approach, (2) The Bridge Approach og (3) The Server Approach. Førstnevnte (4) fordi det skapes et verktøy med sterk flerbrukerbasert sammenkobling mellom deltakerne. Deltakerne kan legge inn musikkmateriale i applikasjonen hvis de har utarbeidet noe nytt som de vil dele med de andre komponistene i systemet. Applikasjonen laster da automatisk opp materialet og sørger for at det blir organisert på samme måte hos alle de andre brukerne.

The Bridge Approach (2) blir benyttet fordi applikasjonen skaper en bro mellom flere samarbeidsdeltakerne der de kan organisere sitt felles musikkmateriale på en strukturert måte uten å være på samme fysiske sted slik dette fenomenet ofte har forekommet i eksisterende praksis. Man kan hevde at dette er, som Weinberg (2002) ytrer, en imitasjon av den eksisterende samarbeidsmåten. Det ble for eksempel nevnt i bakgrunnskapittelet at overlevering av musikkmateriale blir utført både fysisk og digitalt blant intervjuobjektene. I de situasjonene man måtte møtes ansikt til ansikt for å overrekke musikkmateriale, kan man nå bare legge inn det materialet man ønsker å viderebringe til programmet. Når materialet er lastet opp og distribuert til deltakerne kan man videre forklare relevante momenter om musikkmaterialet gjennom synkron og asynkron interaksjon med de andre deltakerne. På denne måten skjer det en imitasjon av den måten gruppesamarbeidet pleide å forløpe seg. For de som brukte e-post til å overlevere musikkmateriale til andre gruppemedlemmer, kan applikasjonen bidra til å forenkle den eksisterende prosessen ved at synkronisering mellom brukerne skjer automatisk. Automatisk synkronisering av deltakerne mot en server medfører også at applikasjonen delvis kan klassifiseres innenfor The Server Approach (3). Hver gang en bruker logges på systemet, synkroniseres brukeren med serveren, og applikasjonen laster automatisk ned aktuelt

musikkmateriale slik at vedkommende kan arbeide med dette lokalt. Serveren sin oppgave (sammen med applikasjonen) ligger altså i automatisk distribusjon av data.

## 7.5 Oppsummering

I dette kapitlet ble det først beskrevet hvordan CMO har blitt utviklet. Dette innebar valg av teknologi, valg av språk, brukerkrav, en fremstilling av den distribuerte klient-/serverarkitekturen, en beskrivelse av databasemodelleringen og systemarkitekturen, og fremgangsmåte for utviklingen ved bruk av en prosessmodell. I tillegg har jeg gjennomgått designvalgene og beskrevet hvordan teori fra CSCW-feltet har blitt reflektert i applikasjonen. Avslutningsvis viste jeg hvordan man kunne anvende Weinbergs (2002) klassifiseringskategorier til å klassifisere CMO.

Samlet utgjør dette kapitlet en potensiell løsning på *hvordan* man kan bruke teori fra CSCW-feltet til å designe en applikasjon for å støtte, forenkle og forbedre distribuert samarbeidende organisering av musikkmateriale. Dette viser imidlertid kun hvordan CSCW er anvendt til å støtte en eksisterende arbeidskontekst; det viser ikke hvorvidt anvendelsen av CSCW til design av applikasjonen er virksom. For å finne ut om applikasjonen fungerer, med hensyn til teori fra CSCW, har det blitt gjennomført en evaluering av applikasjonen. Funn fra evalueringen blir gjennomgått i neste kapittel.

## 8. Evaluering

I dette kapittelet vil jeg presentere funn fra evalueringen av CMO. Tidligere har valget av evalueringsmetoden Groupware Walkthrough (GWW) blitt presentert. Målet med denne metoden er å inspisere brukskvaliteten til gruppevareapplikasjoner ved å måle i hvilken grad applikasjonen støtter samarbeidsmekanismene til Gutwin & Greenberg (2000). GWW ble utført i en laboratoriekontekst der undertegnede deltok som en av brukerne, og evalueringsoppgavene var av både synkron og asynkron karakter. Evalueringen ble, som nevnt, gjennomført i to gjennomganger med to forskjellige brukere, en bruker per gjennomgang.

På de følgende sidene blir det fremlagt 4 scenariobeskrivelser. Dette er beskrivelser som er dannet på bakgrunn av generelle organiseringsaktiviteter som samarbeidende komponister vanligvis har utført. Scenariobeskrivelsene er fiktive, men de er basert på informasjon som ble innhentet fra intervjuene. Hver scenariobeskrivelse har gjennomgått en hierarkisk dekomposisjon, som anbefalt av Pinelle & Gutwin (2002, pp. 457-458), der oppgaver har blitt trukket ut fra beskrivelsen. Disse oppgavene har videre blitt modellert i analysediagrammer som illustrerer ulike måter å løse oppgaven på i CMO. Analysediagrammene inneholder totalt 32 hovedoppgaver med 53 underoppgaver fra alle scenarioene, der hovedoppgavene markerer *hva* som skal utføres, mens underoppgavene viser alternative fremgangsmåter for *hvordan* oppgaven kan utføres. Analysediagrammene ble brukt under den andre hovedkomponenten i GWW: Walkthrough Process (WP). Hver bruker fikk et oppgavesett (i tekstlig format) til hvert scenario basert på oppgavene som var modellert i analysediagrammene, men her ble det kun uttrykt hva som skulle utføres – ikke hvordan det skulle utføres i CMO<sup>45</sup>. Prosjektene og dokumentene som ble brukt på hvert scenario var også nøye beskrevet i oppgavesettene.

Resultatene fra WP er inndelt etter scenario. Til det første scenarioet har jeg inkludert analysediagrammet som ble brukt under gjennomgangsprosessen. Analysediagrammene til de resterende scenarioene er vist i appendiks D.

---

<sup>45</sup> Med unntak av noen oppgaver der det ble brukt tredjehåndsprogramvare.



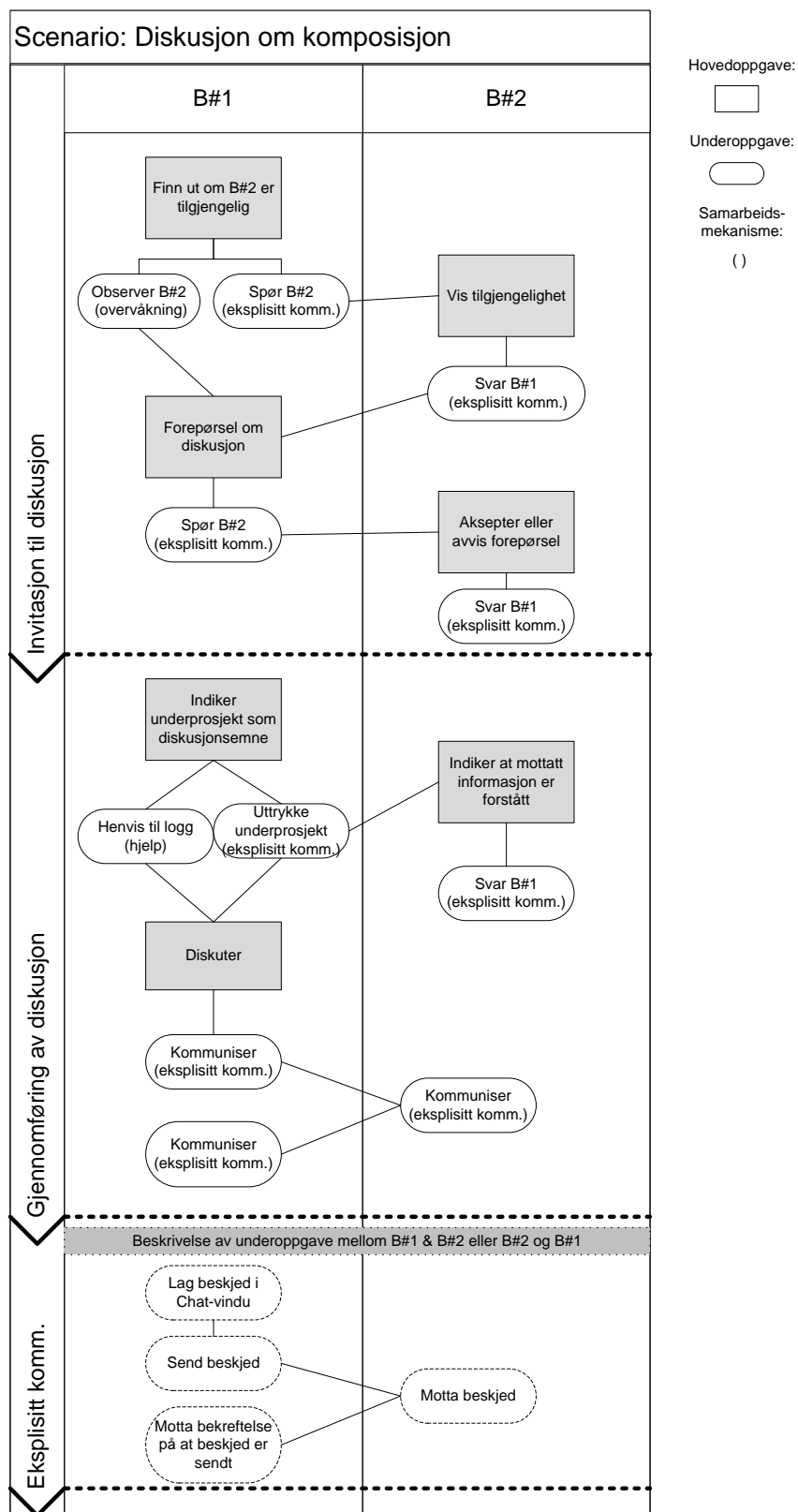
## 8.1 Scenario #1: Diskusjon om komposisjon

Dette scenarioet omhandler samarbeidsaktiviteten «diskusjon om komposisjon». Diskusjoner om musikkmateriale ble, som nevnt tidligere, ansett av Boespflug (1999) for å være en viktig del av den samarbeidende prosessen. Kommunikasjon er en opplagt ingrediens i diskusjoner, og positive resultater fra gjennomgangen av dette scenarioet er derfor av stor betydning. Positive resultater innebærer å demonstrere at CMO har god brukskvalitet som et kommunikasjonsmedium i sammenhenger der diskusjoner mellom deltakerne gjennomføres. I alle scenarioene spilte jeg rollen som B#2.

### 8.1.1 Scenariobeskrivelse

<b>Aktivetsbeskrivelse:</b>	B#1 (gitarist) ønsker å diskutere en komposisjon med B#2 (vokalist). Først må han få tak i B#2, og forhøre seg hvorvidt B#2 har tid til en diskusjon. Etterpå må han spesifisere hvilken komposisjon han ønsker å diskutere, samt forsikre seg om at B#2 er kjent med komposisjonen. Når disse faktorene har blitt klargjort, diskuterer B#1 og B#2 aktuell komposisjon.
<b>Brukerspesifikasjon:</b>	B#1 har stort sett ansvaret for å lage grovskissene til låtene. Han komponerer med gitar og spiller inn komposisjonene i et lydprogram. B#2 skriver tekster og har stor innflytelse på melodilinjene til vokalstemmen. Han må blant annet passe på at vokallinjene er tilpasset hans stemmeregister.
<b>Tilsiktet utfall og resultat:</b>	Utveksle meninger om komposisjonen.
<b>Forhold/omstendigheter:</b>	B#1 og B#2 arbeider fra forskjellige lokasjoner. De bruker e-post til både asynkron kommunikasjon og formidling av komposisjoner. Telefon blir benyttet til synkron kommunikasjon.

### 8.1.2 Analysediagram: Diskusjon om komposisjon



Figur 18 – Analysediagram modellert av scenario #1.

### 8.1.3 Analyse av resultatene

I den første oppgaven må gitaristen (B#1) å finne ut hvorvidt vokalisten (B#2) er tilgjengelig. Dette kan undersøkes på flere måter. For det første må man sjekke om vokalisten er innlogget, og dette kan man raskt observere ved chat-vinduet under innloggede brukere. Etterpå må man finne ut om det er sannsynlig at vokalisten har mulighet til å delta i en diskusjon. Man kan da overvåke vokalisten for å se om han er engasjert i en annen diskusjon med en annen bruker, eller om statusen hans er satt til «away». Et mulig problem med statusfeltet er at dersom man skal kunne stole på denne informasjonen, så er det viktig at brukerne går aktivt inn for å endre på statusen sin når de for eksempel ikke er tilgjengelig. Dersom det viser seg at vokalisten ikke deltar i en annen diskusjon og ikke er borte fra programmet, kan man spørre ham om han har tid til en diskusjon. Den andre underoppgaven går altså ut på å spørre vokalisten om han er tilgjengelig, og dette er et eksempel på samarbeidsmekanismen *eksplisitt kommunikasjon*. Komponentene som inngår i denne underoppgaven er beskrevet nederst i analysediagrammet. Ved å bruke chat-vinduet til eksplisitt kommunikasjon er det avgjørende at begge parter kan opprette og sende meldinger, samt motta en bekreftelse på at den sendte meldingen er mottatt. Dette forløp seg uten problemer. Det var problemfritt å lage meldinger, og man kunne klikke knappen *send* eller *enter-knappen* på tastaturet for å sende meldingene. Serverprogrammet, som mottar meldingene, delegerer dem videre til alle som er medlemmer av gruppen. Dette innebærer at alle direkte meldinger blir vist i chat-vinduet. Både egne og andres meldinger. Og på denne måten får brukerne en bekreftelse på at meldingen har blitt mottatt; de kan se at den meldingen de sendte ble tillagt i chat-vinduet og videre følge diskusjonens kronologiske forløp. Foran hver direkte melding står navnet til avsenderen med tykk font, og dette medførte at det var lett å skille ut de ulike brukernes meldinger.

Majoriteten av oppgavene i dette scenarioet var relatert til eksplisitt kommunikasjon, og det er av den grunn ikke nødvendig å gå i detaljer på hver av dem. Unntaket er oppgaven som heter «indiker underprosjekt som diskusjonsemne». Her kan man enten eksplisitt kommunisere navnet på underprosjektet man ønsker å diskutere eller henvise til loggvinduet om man vil. Hvis gitaristen nettopp har gjort en endring eller lagt til et prosjekt, så kan han uttrykke at han ønsker å diskutere det siste musikkmateriale han la til i CMO. Da har vokalisten mulighet til å sjekke informasjonen i loggvinduet, og automatisk få med informasjon om både hovedprosjekt,

underprosjekt og dokument. På denne måten slipper gitaristen å skrive alt, og samarbeidsmekanismen *hjelp* blir anvendt for å assistere vokalistene i å finne informasjon om diskusjonsemnet gjennom indirekte informasjon som har blitt tillagt i loggvinduet. Under gjennomgangen av CMO valgte den ene brukeren å skrive både hovedprosjekt og underprosjekt i chat-vinduet for å indikere underprosjekt for diskusjonsemne, mens den andre brukeren kun skrev underprosjekt. Dersom man ikke forstår hvilket underprosjekt det er snakk om, er det bare å be brukeren om å spesifisere ytterligere.

Etter at oppgavene var gjennomført ble målekriteriene til Gutwin & Greenberg (2000) gjennomgått på hver oppgave. Det første kriteriet, *effectiveness*, innebar hvorvidt en oppgave ble utført vellykket uten at det inntraff feil i løpet av aktiviteten. Både min og brukerens oppfattelse var at alle oppgavene ble vellykket utført. Ingen feil eller misforståelser inntraff i løpet av gjennomgangen. Begge har erfaring fra direktemeldingsverktøy fra før, og dette kan ha medvirket til at de fleste oppgavene forløp seg uten problemer. Det andre målekriteriet heter *efficiency*, og måler ressursene som ble krevd av brukerne for å utføre oppgavene: et godt gruppevaresystem muliggjør at samarbeidsaktivitetene kan utføres med mindre tid og anstrengelse enn ved et dårlig gruppevaresystem (Gutwin & Greenberg, 2000, p. 100). Det ble ikke brukt mer tid eller anstrengelse enn forventet på hver oppgave. Hvorvidt samarbeidsaktivitetene ble utført med mindre tid og anstrengelse enn det som hadde blitt krevd av et dårlig gruppevaresystem er vanskelig å besvare. Dette kriteriet kan kritiseres for å være upresist. Må man ha en komparativ innfallsvinkel til GWW for å kunne måle samarbeidsmekanismene i forhold til denne måleegenskapen? Umiddelbart kan det se slik ut. Hvis man derimot kun forholder seg til denne måleegenskapen ved å fokusere på ressurser som ble brukt i forhold til hva brukerne forventet å bruke, så opererer man med en mer presis måleegenskap. I resten av studiet vil dette være utgangspunktet for å måle graden av *efficiency*.

Den siste måleegenskapen heter *satisfaction*, og innebærer om man er fornøyd med prosessen og utfallet av oppgavene. Her var begge enig om at alt gikk som forventet og både prosessene og utfallet av oppgavene var tilfredsstillende. Chat-vinduet, som utgjør grunnlaget for eksplisitt kommunikasjon i CMO, var et kjent fenomen for begge brukerne, men den ene brukeren lurte på om man kunne klikke på en bruker i oversikten over innloggede brukere for å sende private

meldinger til ham. Dette går ikke an på nåværende tidspunkt, men bør gjerne implementeres for å la to brukere diskutere privat sammen.

## 8.2 Scenario #2: Endring av komposisjon

I dette scenarioet var målet å kunne endre en komposisjon som allerede var lagt inn i CMO. Endring av komposisjoner er en vanlig aktivitet innenfor samarbeidende organisering. Endringer kan for eksempel bli utført på bakgrunn av at en deltaker har overlevert en komposisjon til en meddeltaker, og at meddeltakerens rolle er å tilføye noe nytt til komposisjonen. Poenget med denne funksjonaliteten er å tilrettelegge for at ikke alle deltakerne trenger å være tilstede samtidig i komposisjonsprosessen, men at enkeltstående deltakere kan endre og legge til instrumenter eller tekst på en komposisjon i ettertid. Dette samsvarer også med empiri fra samarbeidende komponering. Det ble nevnt tidligere at i enkelte ensembler er det ikke nødvendig at alle deltakerne er tilstede under samarbeidet (Boespflug, 1999, p. 35).

### 8.2.1 Scenariobeskrivelse

<b>Aktivitetsbeskrivelse:</b>	En lydtekniker (B#1) ønsker å få tak i den siste komposisjonen som ble endret av gitaristen (B#2) i gruppen han samarbeider med. Hvis endringen har vært avsluttende for komposisjonen, ønsker lydteknikeren å normalisere nivåene på komposisjonen. Når B#1 har fått tak i komposisjonen, må han først identifisere komposisjonsendringen, og etterpå må han også forsikre seg om at B#2 er ferdig å redigere på mediafilen som skal normaliseres. Etter at B#1 har avsluttet normaliseringen av mediafilen, ønsker B#2 å lytte til mediafilen, og tilegne seg kunnskap om hvilken type endring som ble utført av B#1.
<b>Brukerspesifikasjon:</b>	B#1 er en lydtekniker som prosesserer musikkfilene etter at komposisjonene er ferdig. Dette innebærer forskjellige aktiviteter som forbedrer kvaliteten og balansen mellom stemmene på en gitt komposisjon. B#2 er en gitarist som har stor innflytelse på komposisjonene.
<b>Tilsiktet utfall og resultat:</b>	Gjennomføring av en endring på en komposisjon. Begge brukerne må få kunnskap om hverandres handlinger slik at det blir lagt et grunnlag

<b>Forhold/omstendigheter:</b>	for fremtidig samarbeid.
	Kun B#1 og B#2 samarbeider i dette scenarioet. Vanligvis har de formidlet musikkmateriale til hverandre ved bruk av e-post. Kommunikasjon har enten forekommet via e-post eller telefon.

### 8.2.2 Analyse av resultatene

I dette scenarioet er flere samarbeidsmekanismer av vesentlig betydning. Den første oppgaven går ut på at lydteknikeren må finne den siste mediafilen som ble endret av gitaristen. Loggen ble benyttet av begge brukerne for å oppfylle denne oppgaven. Den siste filen som gitaristen endret ble indirekte lagt inn av ham i loggvinduet etter at han hadde endret filen. Alle loggaktivitetene er sortert kronologisk i loggen, der det nyeste bidraget vises først, og av den grunn var det lett for lydteknikeren å identifisere den aktuelle filen. Samarbeidsmekanismen *indirekte kommunikasjon* ble med andre ord tatt i bruk her. Man kan også benytte prosjektutforskeren for å løse denne oppgaven, men dette krever mer tid og anstrengelse av brukeren fordi man må sjekke alle underprosjektene og observere hvilke filer som har gitaristens signatur (det vil si at gitaristens navn er uttrykt i *by user*-feltet). I tillegg må man sammenligne tiden på alle mediafilene for å finne ut hvilken fil som ble lagt til sist. Dersom gitaristen er innlogget er det også en mulighet for lydteknikeren å spørre ham gjennom eksplisitt kommunikasjon i chat-vinduet om hans sist endrede mediafil.

Den neste oppgaven gikk ut på å identifisere hvilken type endring gitaristen har gjort. Lydteknikeren sin rolle (i dette scenarioet) er å normalisere nivåene på volumene i komposisjonen. Det er ikke noe poeng å gjøre dette før gitaristen er ferdig med komposisjonen sin. I versjonsloggen kunne man observere at gitaristen nevnte at hans bidrag til mediafilen var avsluttende, og begge brukerne brukte denne informasjonen for å løse oppgaven. Man kunne også sjekket om det fantes noe på dokumentdetaljene som tilsa at filen var ferdigstilt. Sett at lydteknikeren hadde fulgt komposisjonsprosessen og hadde kunnskap om hva som var planlagt å gjøre før komposisjonen ble avsluttet, så kunne han også lyttet til mediafilen for å sjekke om den var ferdig. Et siste valg er å spørre gitaristen (hvis han er innlogget i CMO) for å få et kontant svar på om han var ferdig.

I den påfølgende oppgaven måtte lydteknikeren sjekke om gitaristen (eller eventuelt andre) redigerte på filen samtidig. Hvis andre hadde redigert på filen samtidig ville dette fått implikasjoner for hvordan filen ble lastet opp til serveren. Siden man ikke kan redigere binærfiler samtidig i sanntid, ville dette medført at (a) en ny fil ble opprettet av det samme dokumentet og videre lastet opp som en ny fil til serveren med lydteknikerens signatur. Og når gitaristen var ferdig med å redigere på samme fil, ville dette resultert i en (b) ny fil på serveren med gitaristens signatur. (Alle tidligere versjoner blir likevel ivaretatt av systemet slik at det er lett å gå tilbake for å endre eller lytte på tidligere versjoner). I gjennomgangen overså den ene brukeren denne oppgaven og hoppet rett til neste oppgave. Den andre brukeren hadde ikke problemer med å identifisere om andre brukere redigerte samme fil.

Neste oppgave gikk ut på å endre filen. Medafilen inneholdt en gitarsolo av gitaristen, og brukeren åpnet og endret filen via Steinberg Wavelab. Etter at filen var lagret, snappet CMO opp endringen, og «Checkout/Upload»-knappen kom til syne. Når brukeren klikket på denne kom versjonsloggen opp med informasjon om at brukeren måtte legge inn informasjon om handlingen han nettopp hadde foretatt seg. Versjonsloggen underbygde i dette henseende samarbeidsmekanismen *planlegging* og *overvåkning*. Brukeren la inn en tekstlig beskrivelse av hva han hadde gjort, og på denne måten kan andre brukere overvåke hva han har gjort, samt planlegge videre fremdrift på bakgrunn av denne informasjonen. Når handlingen var ferdig ble filen automatisk lastet opp til serveren. Men her inntraff en alvorlig feil. Når filen var ferdig lastet opp ble prosjektutforskeren stående i en loop der den oppdaterte vinduet sitt hvert sekund. Dette ble fikset til den andre gjennomgangen, men dette resulterte i at den første brukeren måtte avslutte og starte applikasjonen på nytt. Den andre brukeren erfarte imidlertid ingen problemer med å gjennomføre denne oppgaven.

Etter at lydteknikeren hadde gjennomført endringen var de resterende oppgavenes mål å finne den nyeste endrede mediafilen, samt å identifisere hvilken type endring som var gjort. Dette forløp seg uten problemer. I den første gjennomgangen ble loggvinduet benyttet for å identifisere filen, og versjonsloggen ble benyttet til å finne ut av typen endring. Man kunne også brukt eksplisitt kommunikasjon for å løse disse oppgavene, og dette ble gjort i den andre

gjennomgangen. For å identifisere typen endring så hadde det også vært en mulighet å lytte til mediafilen og sammenligne hvordan den låt før og etter at den ble endret.

Når gjennomgangen var fullført ble måleegenskapene gjennomgått. Med unntak av en oppgave, ble alle oppgavene utført vellykket uten at det inntraff feil. Feilen som inntraff var en typisk bug som tidligere kun har inntruffet en gang. Dette ble som nevnt fikset.

Når det gjelder tid og anstrengelse som ble krevd for å løse oppgavene, viste det seg at når den første brukeren skulle klikke på checkout/upload-knappen måtte han bruke litt tid på å identifisere hvor denne befant seg. Knappen kommer frem og blir vist rett nedenfor prosjektutforskeren når CMO har registrert filendringer. Det kan hende at dette problemet er relatert til manglende brukeropplæring fremfor mangler ved brukergrensesnittet, men det kan også hende at det er en fordel å vise knappen som del av systemet hele tiden. For eksempel ved at knappen er inaktiv (det vil si at den er synlig, men brukeren kan ikke klikke på den) hele tiden i brukergrensesnittet, og at den blir aktiv når brukeren har foretatt en endring som må bekreftes for opplasting. I den andre gjennomgangen ble knappen gjort synlig hele tiden, og denne oppgaven gikk lettere enn før. Det var imidlertid ikke helt intuitivt at man måtte bekrefte opplasting etter at man har lagret en fil, og dette ville vært et viktig moment i en potensiell brukeropplæring. Tanken bak kravet om brukerbekreftelse av alle utførte endringer, er at når en bruker jobber med et dokument, så blir gjerne dokumentet lagret flere ganger underveis i arbeidet. Hvis CMO skulle registrert og opplastet en ny versjon for hver gang brukeren lagret et dokument (uten å overskrive opprinnelsesdokumentet på serveren), ville det blitt mange versjoner å forholde seg til i systemet (og på serveren). At brukeren må bekrefte når han er ferdig å arbeide med et dokument minsker forekomstene av tidligere dokumentversjoner i systemet. Resten av oppgavene ble utført uten at brukeren brukte mer tid og anstrengelse enn forventet.

I forhold til brukertilfredshet var begge generelt fornøyd med oppgavenes utfall, med unntak av oppgaven som resulterte i at brukeren måtte starte applikasjonen på nytt. Det ble også gitt tilstrekkelig informasjon fra systemet etter at hver oppgave ble utført. Den andre brukeren kommenterte imidlertid at also editing-feltet kunne by på misforståelser. Her var det plassert en bindestrek for å indikere at ingen brukere redigerte på det aktuelle dokumentet. Den andre



brukeren lurte på hva denne streken betydde, og det kan tenkes at det ville vært bedre å la feltet være helt tomt.

### 8.3 Scenario #3: Overlevering av musikkmateriale

Det tredje scenarioets mål var å teste hvorvidt CMO har forenklet den eksisterende prosessen som benyttes for å overlevere musikkmateriale fra en deltaker til andre deltakere. Fokuset ligger fremdeles på samarbeidsmekanismene, men dette scenarioet skiller seg litt fra de andre scenarioene ved at det også testes hvorvidt asynkront samarbeid lar seg fasilitere. I de første oppgavene var ikke jeg innlogget, og den andre brukeren måtte derfor identifisere informasjon som var tillagt av meg på forhånd for å kunne løse oppgavene.

#### 8.3.1 Scenariobeskrivelse

<b>Aktivitetsbeskrivelse:</b>	B#1 (gitarist) ønsker å overlevere musikkmateriale til B#2 (bassist). Først må han kontakte B#2 for å vite hvor de skal møtes for å utveksle musikkaterialet. På forhånd må han også vite at det musikkaterialet han overrekker ikke har blitt laget av noen andre medlemmer av gruppen. Når de møtes blir aktuelt musikkmateriale utvekslet.
<b>Brukerspesifikasjon:</b>	B#2 sitt ansvar ligger på melodilinjene og soloene i komposisjonene. Som oftest både vokal og gitar. B#1 er en bassist som ofte lager grovskissene til en låt. Dette innebærer å skape det rytmiske og harmoniske fundamentet for låten.
<b>Tilsiktet utfall og resultat:</b>	Overlevering av musikkmateriale. Formidling av hensikt med musikkmateriale, planlegging av videre progresjon.
<b>Forhold/omstendigheter:</b>	Kun B#1 og B#2 samarbeider i dette scenarioet. Vanligvis utveksler de musikkmateriale mellom hverandre ved å møtes ansikt til ansikt. Da har B#1 mulighet til å formidle tanker omkring komposisjonen, og de har også mulighet til å diskutere videre fremdrift. Telefon blir brukt for å etablere møtested. CD blir vanligvis brukt til oppbevaringsmedium for musikkaterialet.

### 8.3.2 Analyse av resultatene

I den første oppgaven måtte brukeren avgjøre tilgjengeligheten til gitaristen. Dette kunne utføres gjennom samarbeidsmekanismen *overvåkning* ved å se hvem som var innlogget. Den neste oppgaven ba brukeren om å identifisere forslag til videre progresjon på et underprosjekt. Aktuelt hovedprosjekt og underprosjekt var nedskrevet i oppgavesettet hans, men han måtte lete etter tekst som spesifiserte hva som videre burde gjøres på dette underprosjektet. Her kunne man foretatt en egen vurdering av materialet som fantes på underprosjektet i prosjektutforskeren, men dette forutsetter oppdatert kunnskap på alt materialet som finnes der. Da må man med andre ord kjenne til alt som har blitt gjort før, og vite hva som vanligvis er behovet på en komposisjon på et gitt steg i komposisjonsprosessen. Et annet alternativ er å benytte kommentarvinduet til å se på tidligere tekstinnlegg fra andre brukere. Her hadde det blitt lagt til tekst på forhånd om at rytmisk grovskisse og vokalstemme var ferdig, men at komposisjonen trengte en gitarsolo. I den neste oppgaven måtte brukeren velge å legge til musikkmateriale på bakgrunn av informasjonen han hadde tilegnet seg. I oppgavesettet fikk han tre alternativer til filer han kunne velge fra: rytmisk grovskisse, vokalstemme og gitarsolo. Begge brukerne hadde klart å lese i kommentarvinduet at det fantes et behov for en gitarsolo, og valgte riktig alternativ her. Men når den første brukeren skulle legge til aktuell fil benyttet han ikke *add file*-knappen. Han klikket i stedet på *open*-knappen, og skjønnte at noe var galt når filen bare åpnet seg. Han fikk beskjed av meg om at *add file*-knappen var det korrekte valget, og la til aktuelt musikkmateriale. Den andre brukeren klarte imidlertid å velge *add file*-knappen, men lurte på hva som var hensikten med tittel- og beskrivelsesfeltene som kom frem når man hadde klikket *add file*-knappen.

I den neste oppgaven ble brukeren bedt om å formidle forslag til videre fremdrift på komposisjonen. Dette kommer inn under samarbeidsmekanismen *planlegging*, og det var tenkt at brukeren kunne enten oppdatere dokumentdetaljene for å illustrere dette eller legge inn et nytt innlegg i kommentarvinduet. Den første brukeren virket litt usikker på hvilken av disse som var mest hensiktsmessig å anvende, men valget falt på kommentarvinduet. Den andre brukeren benyttet også kommentarvinduet. Forslag til videre progresjon kunne de finne på selv, men det var gitt noen eksempler i oppgavesettet deres. Når innlegget var lagt inn, logget undertegnende seg inn i applikasjonen for å løse oppgavene som var gitt til bassisten (B#2) i scenarioet. De resterende oppgavene gikk ut på å identifisere tillagt musikkmateriale, samt finne ut om det

fantas en plan for videre progresjon. Her ble loggen anvendt for å identifisere tillagt musikkmateriale, og jeg (som bruker B#2) spurte den første brukeren om plan for videre progresjon gjennom chat-vinduet. Her ble jeg henvist til å sjekke ut kommentarvinduet. Dette er et alternativ som jeg ikke hadde modellert i analysediagrammene, og hører sammen med samarbeidsmekanismen *hjelp*, der målet er at deltakere skal kunne tilby hjelp til hverandre for å illustrere hva og hvor de er i arbeidet sitt. Grunnlaget for hjelpesamarbeidsmekanismen bygger på eksplisitt kommunikasjon (se seksjon 3.6 om samarbeidsmekanismene). I den andre gjennomgangen gikk jeg direkte til kommentarvinduet for å se om det var lagt inn en plan der. Her kunne jeg se at den andre brukeren hadde lagt inn et nytt kommentarinlegg med forslag til videre progresjon.

I henhold til måleegenskapen effectiveness ble majoriteten av oppgavene utført vellykket uten feil. At den ene brukeren valgte *open*-knappen i stedet for *add file*-knappen kunne muligens blitt forbedret ved å plassere *add file*-knappen før *open*-knappen i listen over *File Controls* (jf. skjermbildet i seksjon 7.3.3). Eventuelt kunne man endret navn på *open*-knappen til *edit* for å indikere at man bruker knappen til redigering. Dette aspektet kan imidlertid være relatert til manglende brukeropplæring.

De fleste oppgavene ble utført uten mer tid og anstrengelse enn forventet (efficiency). Men i oppgaven der brukeren skulle legge inn forslag til videre progresjon brukte brukeren litt tid på å finne ut om han burde oppdatere dokumentdetaljene eller legge inn et tekstbidrag i kommentarvinduet. Forholdet mellom dokumentdetaljer og kommentarvinduet ble altså ikke oppfattet som intuitivt av brukeren. Når man legger inn et dokument i CMO må dokumentdetaljene tittel og beskrivelse legges inn for å beskrive dokumentet (jf. skjermbildet i seksjon 7.3.3). Det er også muligheter for å endre dette i ettertid. Det kan tenkes at muligheten for å endre denne informasjonen er overflødig, og at det hadde vært nok å legge dette inn en gang. Men samtidig kan det være uheldig å begrense brukernes muligheter til å rette opp i eventuelle dokumentdetaljer som ble skrevet inn feil eller som ble oppfattet av andre brukere som mangelfulle. Eventuelt kunne man presentert en informasjonsboks til brukeren når han nettopp hadde lagt til et dokument der programmet tipset ham om å legge inn et kommentarinlegg i kommentarvinduet for å gi et forslag til videre progresjon. I den andre

gjennomgangen ble brukeren usikker på hva nettopp feltene tittel og beskrivelse betydde i vinduet (dokumentdetaljene) som kom opp når han klikket på *add file*-knappen. Her burde det muligens vært skrevet hvilken informasjon som var ønsket, samt hva som var hensikten med informasjonen.

Måleegenskapen satisfaction ble generelt tilfredsstilt. Begge var fornøyd med utfallet av de fleste oppgavene. Men når den første brukeren hadde lagt inn et tekstbidrag i kommentarvinduet under tabulatoren «New entry» på kommentarvinduet (jf. skjermbildet i seksjon 7.3.5), mente han at programmet burde ha skiftet tilbake til kommentarene etter at han hadde klikket på *save entry*. CMO presenterer riktignok informasjon om at innlegget ble lagret, men det kunne tenkes at brukertilfredsheten ville blitt forbedret hvis CMO i tillegg automatiserte prosessen ved å gå tilbake til oversikten over kommentarer i kommentarvinduet. Den andre brukeren klikket seg imidlertid rett tilbake til kommentarvinduet for å sjekke om innlegget hans ble lagt inn.

## 8.4 Scenario #4: Arrangere øving

Det siste scenarioet går litt utover det som tidligere har blitt omtalt som organiseringsaktiviteter. Samarbeidsmekanismene er av stor betydning her, men målet var å finne ut hvorvidt CMO kan anvendes til å arrangere øving mellom to eller flere deltakere. Awareness om hvor brukerne er når de benytter applikasjonen var også viktig i dette scenarioet.

### 8.4.1 Scenariobeskrivelse

<b>Aktivitetsbeskrivelse:</b>	B#1 (bassist) ønsker å arrangere en øving med B#2 (trommeslager). Først må han få tak i B#2, og forhøre seg hvorvidt B#2 er i byen, og om B#2 har tid til øving. Når han får tak i B#2 må de avgjøre en dato og tidspunkt for øving.
<b>Brukerspesifikasjon:</b>	B#1 bor i Bergen. Han er medlem av et pop-band sammen med trommeslageren. Begge deler øvingslokale. B#2 jobber imidlertid frilans og er ofte utenbys.
<b>Tilsiktet utfall og resultat:</b>	Oppnå kontakt og avgjøre tidspunkt for øving.

<b>Forhold/omstendigheter:</b>	B#1 og B#2 øver og komponerer sammen. Fokuset ligger på å lage rytmiske fundamenter for fremtidige komposisjoner. De trenger ikke resten av bandet for å gjøre denne aktiviteten. Telefon har blitt benyttet til synkron kommunikasjon for å avtale øvinger.
--------------------------------	--

### 8.4.2 Analyse av resultatene

I den første oppgaven måtte brukerne logge inn i CMO. Her måtte også brukerne spesifisere hvor de jobbet fra. I den neste oppgaven måtte de *overvåke* og observere *indirekte informasjon* avgitt fra den andre brukeren. Er han innlogget? Viser statusen hans at han er tilgjengelig? Hvor arbeider han fra? Dette var viktige spørsmål som brukeren måtte stille seg selv her. Oversikten over brukere i chat-vinduet (jf. skjermbildet i seksjon 7.3.4) viste både informasjon om status og hvor den andre brukeren befant seg. Denne informasjonen ble brukt for å svare på oppgaven, men brukeren kunne også spurt direkte gjennom *eksplisitt kommunikasjon* i chat-vinduet. I oppgavesettet ble brukeren videre bedt om å finne ut hvorvidt hans meddeltaker befant seg innenfor samme geografiske område, og at han måtte bruke denne informasjonen til å avgjøre om det var sannsynlig at hans meddeltaker kunne være med på en øving. Dersom det hadde vist seg at de ikke befant seg i samme land, og at trommeslageren (B#2 i dette scenarioet) var på turne, så hadde det vært poengløst å spørre om øving neste dag. Man kunne imidlertid forhørt seg om tidspunkt for trommeslagerens retur til hjemstedet hans. (Hjemsted er inkludert i brukerinformasjonsboksen dersom man høyreklikker på brukerne. Se seksjon 7.3.4). Den siste oppgaven gikk ut på å formidle en forespørsel om øvingstidspunkt til den andre brukeren. Eksplisitt kommunikasjon ble brukt til å løse oppgaven.

Oppgavene ble utført vellykket uten at det inntraff feil i løpet av gjennomgangen av dette scenarioet. Det ble heller ikke brukt mer tid og anstrengelse enn forventet på å løse oppgavene. Når det gjelder måleegenskapen *satisfaction* var ikke brukerne helt fornøyd med brukerinformasjonsboksen som kom opp når man høyreklikket på brukeren. Denne hadde ikke lukkekryss, og det var ikke intuitivt at man bare kunne klikke et annet sted i applikasjonen for å få den bort. Lukkekryss burde med andre ord vært tilstede. I tillegg kommenterte den andre brukeren at det ikke var intuitivt at man skulle høyreklikke på brukerne for å få opp

brukerinformasjon om dem. Når det gjelder brukerinformasjonen så kunne man gjort det slik at når man klikket med venstre museknapp på en gitt bruker, så kom brukerinformasjonen opp, men dette kan også være relatert til manglende brukeropplæring. Designvalget er basert på modellen fra desktopprogrammer (som for eksempel Windows) der man må høyreklikke for å få opp egenskaper til et gitt objekt.

## 8.5 Resultater fra Groupware Walkthrough

Resultatene fra GWW avslørte noen problemer med brukskvaliteten til CMO. Dataen fra evalueringen ble samlet inn gjennom observasjon og etterfølgende spørsmål om hver av oppgavene. Spørsmålene var designet for å måle effectiveness, efficiency og satisfaction ved samarbeidsmekanismene. Noen av problemene var ikke direkte relatert til samarbeidsmekanismene, og kunne antageligvis blitt løst bedre gjennom anvendelse av alternative evalueringsrammeverk. Andre problemer ble forsøkt løst gjennom endringer i programmeringskoden mellom de to gjennomgangene.

Målet med GWW er som nevnt å måle brukskvaliteten i et gruppevaresystem, og groupware-brukskvalitet er definert av Gutwin & Greenberg (2000) som: i hvilken grad et gruppevaresystem støtter samarbeidsmekanismene for et gitt sett av brukere og et gitt sett av oppgaver (se seksjon 6.3).

Man kan benytte en tabell for å sette opp hver samarbeidsmekanisme mot hver måleegenskap (Gutwin & Greenberg, 2000, p. 100). I tabell 5 vises en tabell av denne typen. Her er det illustrert en oversikt over alle samarbeidsmekanismene og måleegenskapene, samt i hvilken grad samarbeidsmekanismene er støttet.

Tabell 5 – I hvilken grad samarbeidsmekanismene er støttet i Collaborative Music Organizer.

	Effectiveness	Efficiency	Satisfaction
<b>Eksplisitt kommunikasjon:</b>	Ja	Ja	Ja
<b>Indirekte kommunikasjon:</b>	Feil inntraff ved endring av komposisjon, noe som gjorde at indirekte informasjon ikke ble innlagt i loggen. Feilen ble imidlertid fikset til den andre gjennomgangen.	Ja	Ja
<b>Koordinasjon av handling:</b>	Ja	Noe uklar sammenheng mellom dokumentdetaljer og kommentarvindu.	Ja, men noe usikkerhet preget en av oppgavene.
<b>Planlegging:</b>	Ja	Ja	Hovedsakelig ja, men automatisk tilbakegang til kommentarvinduet ble etterspurt
<b>Overvåkning:</b>	Ja	Ja	Mesteparten av funksjonaliteten var tilfredsstillende, men det ble registrert misnøye med brukerinformasjonsboksen.
<b>Hjelp:</b>	Ja	Ja	Ja
<b>Beskyttelse:</b>	Ja (men manglende funksjonalitet for sletting av prosjekter)	Ja	Ja

Noen av feilene eller manglene kan ikke direkte relateres til samarbeidsmekanismene og har av den grunn ikke blitt inkludert i tabellen, men konsekvensen av feilen kan relateres. I scenarioet der brukeren skulle endre en komposisjon ble prosjektutforskeren stående og oppdatere seg selv etter bekreftet checkout/upload. Denne feilen resulterte i at loggen ikke ble oppdatert med informasjon om endringen av komposisjonen. Grunnlaget for oppdatering av loggen er basert på indirekte kommunikasjon fra deltakerne. Av den grunn kan man hevde at samarbeidsmekanismen indirekte kommunikasjon ikke fungerte optimalt i CMO i den første gjennomgangen.

I dette kapitlet har jeg gjennomgått evalueringen av CMO. Her ble det vist hvordan GWW ble anvendt for å inspisere gruppevarebrukskvaliteten i applikasjonen, og noen problemer ble avdekket. I neste kapittel skal jeg konkludere og evaluere studiet, samt presentere forslag til videre forskning.



## 9. Konklusjon

Dette studiet har fokusert på utviklingen og evalueringen av Collaborative Music Organizer (CMO). Utviklingen har blitt gjennomført med basis i teori fra CSCW, og både bruk av teori og designvalg er dokumentert i kapittel 7. Målet med dette studiet er å tilby en IKT-løsning som kan bidra til å støtte samarbeidende komponister i organiseringen av deres felles musikkmateriale. Utgangspunktet for konstruksjon av løsningen er basert på forskningslitteratur og intervjuer med aktører innenfor flere typer musikkvirksomhet.

I det første forskningsspørsmålet ble det spurt hvordan samarbeidende organisering av musikkmateriale foregår i eksisterende praksis. Dette ble besvart gjennom intervjuer med flere aktører som var aktiv innenfor musikkkyrket, og som hadde erfaring fra samarbeidende komposisjon. I det andre forskningsspørsmålet ble det spurt hvordan CSCW kan benyttes til å forbedre samarbeidende organisering av musikkmateriale. Ved å designe applikasjonen slik at funksjonalitet og brukergrensesnitt reflekterer teori fra CSCW-feltet, har man forsøkt å tilby et potensielt svar på det andre forskningsspørsmålet. Ved å vise for eksempel hvordan applikasjonen underbygger samarbeidsmekanismene, har man vist hvordan man kan bruke CSCW til å støtte det musikalske samarbeidet. Alle samarbeidsmekanismene har blitt dekket gjennom presentasjonen av designvalgene og gjennom funnene fra evalueringen, og disse representerer funksjonalitet som bør være tilgjengelig i et shared workspace. Hensynet til awareness har blitt gjenspeilet i applikasjonen gjentatte ganger ved å vise hva andre deltakere gjør på, hvor de er, hva de har gjort, om de er tilgjengelig og så videre. Sentrale momenter innenfor Common Information Spaces (CIS) har blitt dekket, eksempelvis gjennom å tilby muligheter for at deltakerne kan diskutere intensjonen med informasjonsobjektene sine, og på den måten øke sin forståelse om konteksten som informasjonsobjektet kom fra. I tillegg er det nevnt at å konstruere et CIS er en god fremgangsmåte for å støtte articulation work.

Applikasjonen tilrettelegger også for at ulike brukere med forskjellig bakgrunn og kompetanse kan bli bevisst på sine meddeltakeres bakgrunn og kompetanse. På denne måten forbedrer man grunnlaget for at deltakerne kan oppnå en felles forståelse av informasjonsobjektene.

En vesentlig del av det andre forskningsspørsmålet har altså blitt besvart gjennom å bruke teori fra CSCW. Men når man snakker om forenkling og forbedring gjennom å bruke teori fra CSCW-

feltet, så er det viktig også å vurdere dette i relasjon til de problemene som ble identifisert innenfor den eksisterende arbeidskonteksten. Den eksisterende praksisen for samarbeidende organisering var relativt ad hoc, og bestod av mange forskjellige fremgangsmåter for overlevering og oppbevaring av musikkmateriale, samt valg av kommunikasjonsmedium. Dette kunne (og hadde) ledet til flere problemer: musikalsk arbeid ble glemt, komposisjoner forsvant, det var lett å ekskludere deltakere fra overleveringen av en komposisjon, det kunne kreves mye ekstraarbeid for å oppdatere deltakere på et musikalsk prosjekt som ikke har blitt utviklet kontinuerlig, og fysiske innspillingsmedier kunne bli ødelagt.

CMO har blitt utviklet med mål om å kunne forbedre eller forenkle det eksisterende samarbeidet. Dette innebærer å løse de overnevnte problemene og utfordringene. Det er imidlertid en forutsetning at brukerne har datamaskin- og internetttilgang, og at de har CMO installert for at CMO kan anvendes til å støtte det eksisterende samarbeidet. Når man snakker om hvorvidt CMO har bidratt til å forbedre eller forenkle samarbeidet, er det også en forutsetning at CMO fungerer optimalt. Men man kan aldri garantere for at system- og programfeil ikke vil inntreffe, og den følgende argumentasjonen må derfor betraktes i forhold til potensialet ved å bruke CMO.

En av intensjonene med utviklingen av CMO er å bidra til å strukturere den eksisterende praksisen (og alle fremgangsmåtene som var i bruk her) ved å tilby et enkeltstående felles system for organiseringen av musikkmaterialet slik at man slipper å benytte mange forskjellige fremgangsmåter i organiseringen. Dette kan medvirke til å forenkle det eksisterende samarbeidet. I tillegg er CMO sterkt tilknyttet en sentralisert server som oppbevarer og synkroniserer relevant musikkmateriale mellom deltakerne, og dette kan medvirke til å forbedre samarbeidsforholdene siden deltakerne ikke lenger trenger å frykte at data skal gå tapt. Samtidig kan det være enklere å overlevere musikkmateriale til andre deltakere ved bruk av CMO siden overleveringen blir automatisert når man har lagt til nytt musikkmateriale i systemet. Mens det tidligere var lett å ekskludere noen deltakere fra overleveringen av nytt eller endret musikkmateriale, er det nå tilrettelagt for at alle som deltar på en gruppe kan orientere seg om nytt og endret musikkmateriale som er relevant for dem; dette gjelder selv om ikke alle benyttet applikasjonen under innleggelsen av musikkmateriale. I tillegg trenger ikke en deltaker å etterspørre musikkmateriale som ble utvekslet mellom andre meddeltakere uten at han var tilstede. En

deltaker trenger heller ikke å etterspørre musikkmateriale dersom han selv har skyld i at det forsvant eller gikk tapt.

Formålet med evalueringen var å finne ut hvorvidt CMO hadde god gruppevarebrukskvalitet gjennom å måle hvorvidt applikasjonen støttet samarbeidsmekanismene til Gutwin & Greenberg (2000). Generelt viste evalueringsgjennomgangene tilfredsstillende resultater, selv om ikke grunnlaget for alle samarbeidsmekanismene fungerte optimalt. Forslag til noen forbedringer og alternative løsninger ble derfor presentert under analysen.

I lys av argumentasjonen ovenfor kan man betrakte CMO som et fruktbart utgangspunkt for å assistere samarbeidende komponister gjennom organiseringsaktivitetene deres. I neste seksjon skal jeg presentere en kort evaluering av studiet, og avslutningsvis vil det presenteres forslag til videre forskning.

## **9.1 Evaluering av studiet**

Det er klart at både funnene som ble innsamlet fra intervjuene og designvalgene bak CMO har sine begrensninger. Når det gjelder funnene fra intervjuene, kan man ikke generalisere disse til å gjelde uavhengig av populasjonsstørrelse. Man kan heller ikke utarbeide statistikk på bakgrunn av dem. For å gjøre generaliserbare studier om den eksisterende praksisen, som er gjeldende uavhengig av populasjonsstørrelse, burde man hatt et større antall komponister som utgangspunkt for datainnsamlingen. Dette ville ført til et rikere sett av datamateriale.

Evalueringen ble utført i en laboratoriekontekst med et begrenset mål. Målet var å finne ut hvorvidt applikasjonen hadde god gruppevarebrukskvalitet gjennom å måle om applikasjonen underbygget samarbeidsmekanismene. Dette er en klar begrensning, og man vil ikke nødvendigvis få kartlagt om eksempelvis navigasjonen fungerer bra, om den generelle forståelsen av brukergrensesnittet er tilfredsstillende, eller om applikasjonen er robust over lengre tid med last fra mange brukere. I kapittelet om evalueringsfunnene ble det for eksempel nevnt problemer med å få forstå forholdet mellom dokumentdetaljene og kommentarvinduet. Brukerne stilte spørsmål til hensikten med både dokumentdetaljene og kommentarvinduet, og i hvilke situasjoner de skulle foretrekke å bruke den ene fremfor den andre. Denne typen spørsmål

kunne blitt avdekket tidligere, men da måtte man gjerne ha utført andre evalueringer før Groupware Walkthrough ble effektuert<sup>46</sup>.

Det ble nevnt tidligere at det var gjort noen antagelser angående hvilke områder som var viktig for å undersøke samarbeidende organisering av musikkmateriale i eksisterende praksis (se seksjon 2.3). Antagelsene var delvis basert på forskningslitteratur og delvis basert på lang personlig erfaring innenfor et bredt spekter av musikkvirksomhet. Den personlige erfaringen innebærer også, som nevnt, kjennskap til de fleste av intervjuobjektene. Før selve studiet ble påbegynt hadde undertegnede erfaring allerede medvirket til å danne en forståelse av hvordan samarbeidende organisering foregår. Med slike omstendigheter er det på sin plass å ta en kort diskusjon om forskerbias.

Forskerrollen bør være upartisk og strebe etter nøytralitet. Når forskeren i dette studiet har tidligere erfaring fra denne praksisen, kan dette kalles *forlenget engasjement*<sup>47</sup> (Robson, 2002, pp. 172-174). Forlenget engasjement er et begrep som brukes i tilknytning til etnografiske studier når en forsker eksempelvis befinner seg sammen med de forskingsobjektene han skal undersøke over lengre tid. Dette innebærer at forskerens forståelse av området han forsker på blir påvirket av menneskene rundt ham (a) når de blir vant til forskerens tilstedeværelse og (b) når forskeren blir vant til menneskene rundt ham. Man bør være bevisst på at dette påvirker antagelsene og forståelsen man har av fenomenet som skal undersøkes. Men samtidig er alle antagelsene i dette studiet fulgt opp gjennom intervjuer, og av den grunn kan man anta at man har fått innsikt i den reelle musikkpraksisen. Fordelene når forskeren kjenner noen av intervjuobjektene, er at intervjuobjektene kan ha en mer ærlig holdning til forskeren (*ibid*).

I dette studiet har kjennskapen til musikkpraksisen også vært avgjørende for å identifisere utgangspunktet for studiet. Uten tidligere personlig erfaring fra dette området ville det vært relativt vanskelig å tilegne seg kunnskap om problemene som eksisterer innenfor samarbeidende organisering av musikkmateriale. Uten personlig erfaring ville man med andre ord hatt mye

---

<sup>46</sup> Se Pinelle & Gutwin (2000) for en oversikt over alternative evalueringsmetoder for groupware.

<sup>47</sup> Eng.: prolonged involvement.

mindre kunnskap som igjen kunne resultert i at problemstillingene ikke hadde blitt identifisert. Følgende ville man også vært svakere skikket til å undersøke musikkpraksisen ytterligere.

## **9.2 Videre forskning og fremtidige utfordringer**

Evalueringen viste at CMO hadde relativt god gruppebrukskvalitet, men noen problemer ble imidlertid identifisert. Disse problemene, samt begrensningene som ble nevnt tidligere i kapittelet, kan bidra til å skape nye problemstillinger innenfor problemområdet til dette studiet. Dette kan være aktuelle utgangspunkt for fremtidig forskning.

Tidligere ble det nevnt hvilke kommunikasjonsmedier som ble anvendt i det eksisterende samarbeidet. Telefon og e-post var to eksempler på dette. I fremtidige studier kunne man derfor videreutviklet CMO ved å implementere voice chat og e-postfunksjonalitet slik at man i større grad tilrettelegger for at samarbeidet kan få utfolde seg også slik det forekommer i eksisterende praksis. For eksempel ved å sende en e-post til brukerne hver gang et nytt prosjekt ble opprettet eller nytt musikkmateriale ble tillagt. I tillegg burde man hatt mulighet til å sende private meldinger til andre brukere gjennom chat-vinduet. Dette ble som nevnt etterspurt under evalueringen av CMO.

CMO sin innloggingsfunksjonalitet begrenset brukerne i å være medlemmer av flere grupper. Det er problemfritt å knytte en bruker mot flere grupper i databasen, men dette har ikke blitt gjenspeilet i innloggingsprosedyren. Fraværet av denne funksjonaliteten hadde ingen implikasjoner for å teste samarbeidsmekanismene gjennom evalueringen av CMO, men i eventuelle fremtidige studier kunne dette vært et ekstravalg; det vil si at brukerne får beskjed om å velge gruppetilhørighet når de logger seg inn.

Forholdet mellom rollene til musikere, lydteknikere og produsenter har blitt forenklet i dette studiet. Det har ikke blitt tatt spesielt hensyn til hvordan disse tre rollene skiller seg i forhold til rollenes spesifikke arbeidsoppgaver, og fremtidige studier kunne derfor analysert disse forholdene slik at applikasjonen kunne tilpasses hver rolle. Rollene har gjerne fellestrekk, men det kunne vært interessant å finne ut hvor de skiller seg innenfor samarbeidet. I tillegg kunne man i et fremtidig forskningsprosjekt videreutviklet CMO til å støtte musikere, komponister,

lydteknikere og produsenter fra begynnelsen av samarbeidet deres, til komposisjonene er ferdig og klar for distribusjon. For eksempel ved å integrere muligheter for opplasting av ferdige komposisjoner direkte til webbutikker som selger musikkfiler.

En aktuell problemstilling for fremtidig forskning er også å undersøke hvorvidt CMO kan fungere i en reell arbeidskontekst med samarbeidende komponister uten at scenarioer har blitt modellert på forhånd. Det hadde vært interessant å observere om brukerne erfarte at applikasjonen støttet samarbeidet deres.

CSCW har vært et vesentlig forskningsfelt for dette studiet, og mye av forskningslitteraturen som er brukt i studiet har blitt publisert i relasjon til dette feltet. I tillegg har det blitt presentert forskningslitteratur med fokus på musikk. Musikk-literaturen har ikke nødvendigvis blitt identifisert med et avgrenset forskningsfelt, men har blitt publisert i journaler som omfatter forskningslitteratur om kunst, estetikk, musikk-sosiologi, musikk-historie, læring og musikk-pedagogikk. Det finnes derimot andre forskningsfelt som kunne vært interessant i fremtidig forskning innenfor dette studiets problemområde. Et av disse feltene kalles *ethnomusicology*, og er et interdisiplinært forskningsfelt som er opptatt av menneskelige musikkaktiviteter fra hele verden. Dette inkluderer studier på musikk, menneskene som lager musikken, instrumentene de bruker, og prosessene som er involvert (The Society for Ethnomusicology, 2008). Dette feltet kunne muligens bidratt med interessant informasjon om for eksempel forholdet mellom de ulike rollene, det vil si musikere, lydteknikere og produsenter, som er involvert i musikkproduksjon.

I dette studiet har jeg vist hvordan man kan bruke CSCW til å designe og evaluere en applikasjon med mål om å støtte distribuert samarbeidende organisering av musikkmateriale, og jeg håper at dette studiet kan være et interessant bidrag til CSCW-feltet.

## Litteraturliste

- Alperson, P. (2008). The instrumentality of Music. *The Journal of Aesthetics and Art Criticism*, 66(1), 37-51.
- Andersen, S., & Abella, V. (2004). Changes to Functionality in Microsoft Windows XP Service Pack 2 (Part 5: Enhanced Browsing Security). Retrieved 20.04 2008, from <http://technet.microsoft.com/en-us/library/bb457150.aspx#EHAA>
- Association for Information Systems. (2007). Design Research in Information Systems (Research). Retrieved 18.03 2007, from <http://isworld.org>
- Bannon, L. (2000). Understanding Common Information Spaces in CSCW, *Workshop on Cooperative Organisation of Common Information Spaces*. Copenhagen, Denmark.
- Bannon, L., & Bødker, S. (1997). *Constructing Common Information Spaces*. Proceedings of the fifth conference on European Conference on Computer Supported Cooperative Work, Lancaster, UK.
- Bannon, L., & Robinson, M. (1991). *Questioning representations*. Proceedings of the second conference on European Conference on Computer-Supported Cooperative Work, Amsterdam, The Netherlands.
- Bannon, L., & Schmidt, K. (1989). CSCW: Four Characters in Search of a Context. *Proceedings of the First European Conference on Computer Supported Cooperative Work*, Gatwick, London.
- Barbosa, A. (2003). Displaced Soundscapes: A survey of Network Systems for Music and Sonic Art Creation. *Leonardo Music Journal*, 13, 53-59.
- Barbosa, A. (2006). *Computer-supported cooperative work for music applications*. Unpublished Ph.D. Dissertation, UPF, Barcelona.
- Bellini, P., Fioravanti, F., & Nesi, P. (1999). Managing music in orchestras. *Computer*, 32(9), 26-34.
- Bennet, S. (1976). The Process of Musical Creation: Interviews with Eight Composers. *Journal of Research in Music Education*, 24(1), 3-13.
- Boespflug, G. (1999). Popular Music and the Instrumental Ensemble. *Music Educators Journal*, 85(6), 33-37.
- Digital Audio Working Group. (2005). Digital Audio Best Practices. Retrieved 24.02 2008, from <http://www.bcr.org/cdp/best/digital-audio-bp.pdf>
- Dourish, P., & Bellotti, V. (1992). *Awareness and coordination in shared workspaces*. Proceedings of the 1992 ACM conference on Computer-supported cooperative work, Toronto, Ontario, Canada.

- Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). Groupware: Some issues and experiences. *Communications of the ACM*, 34(1), 39-58.
- Elmasri, R., & Navathe, S. B. (2003). *Fundamentals of Database Systems* (4 ed.). Boston, MA: Addison Wesley.
- Fjuk, A., Smørðal, O., & Nurminen, M. I. (1997). Taking articulation work seriously - an activity theoretical approach, *Technical Report: TUCS-TR-120*. Turku Centre for Computer Science.
- Gang, D., Chockler, V., Anker, T., Kremer, A., & Winkler, T. (1997). *TransMIDI: A System for MIDI Sessions Over the Network Using Transis*. Proceedings of the International Computer Music Conference, Thessaloniki, Greece.
- Gerson, E. M., & Star, S. L. (1986). Analyzing due process in the workplace. *ACM Transactions on Office Information Systems*, 4(3), 257-270.
- Greif, I. (I. Greif) (1988). *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo, California: Morgan Kaufmann Publishers.
- Grimen, H. (K. O. Amås) (2001). *Samfunnsvitenskaplige tenkemåter* (2. ed.). Oslo: Universitetsforlaget.
- Gutwin, C., & Greenberg, S. (1999). The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 6(3), 243-281.
- Gutwin, C., & Greenberg, S. (2000). *The Mechanics of Collaboration: Developing low cost usability evaluation methods for shared workspaces*. Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Gaithersburg, MD.
- Gutwin, C., & Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work* 11(3), 411-446.
- Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS QUART*, 28(1), 75-105.
- Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., et al. (1996). ACM SIGCHI Curricula for Human-Computer Interaction. Retrieved 27.03 2008, from <http://sigchi.org/cdg/index.html>
- Johansen, R. (1988). *GroupWare: Computer Support for Business Teams*. New York, USA: The Free Press
- Jordà, S. (2000). Faust Music On Line (FMOL): An approach to Real-time Collective Composition on the Internet. *Leonardo Music Journal* 9, 5-12.
- Koch, M., & Gross, T. (2006). *Computer-Supported Cooperative Work - Concepts and Trends*. Proceedings of the 11th Conference of the Association Information and Management (AIM), Luxembourg, Luxembourg.



- Kunnskapsforlaget. (1975-2008). Norsk Riksmålsordbok. Retrieved 09.02 2008, from [http://ordnett.no/ordbok.html?search=organisere&search\\_type=&publications=6](http://ordnett.no/ordbok.html?search=organisere&search_type=&publications=6)
- Latta, C. (1991). Notes from the NetJam Project. *Leonardo Music Journal*, 1(1), 103-105.
- Lewis, G. H. (1988). The creation of popular music: A comparison of the "Art Worlds" of American Country Music and British Punk. *International review of the Aesthetics and Sociology of Music*, 19(1), 35-51.
- Mozilla. (2008). Links to local or network pages do not work. Retrieved 20.04 2008, from <http://support.mozilla.com/en-US/kb/Links+to+local+or+network+pages+do+not+work>
- Pazel, D., Abrams, S., Fuhrer, R., Oppenheim, D., & Wright, J. (2000). *A Distributed Interactive Music Application Using Harmonic Constraint*. Proceedings of the 2000 International Computer Music Conference, Berlin, Germany.
- Pinelle, D., & Gutwin, C. (2000). *A Review of Groupware Evaluations*. Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Gaithersburg, Maryland, USA.
- Pinelle, D., & Gutwin, C. (2001). *Group Task Analysis for Groupware Usability Evaluations*. Proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Cambridge, Massachusetts, USA.
- Pinelle, D., & Gutwin, C. (2002). *Groupware walkthrough: Adding context to groupware usability evaluation*. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, Minneapolis, Minnesota, USA.
- Randell, D. M. (1987). Defining music. *Notes, Second Series*, 43(4), 751-766.
- Robson, C. (2002). *Real world research* (2 ed.). Oxford, UK: Blackwell Publishing.
- Rodden, T. (1991). A Survey of CSCW Systems. *Interacting with computers - the interdisciplinary journal of human-computer interaction*, 3(3), 319-353.
- Rubinstein, R. (1995). Computer science projects with music. *ACM SIGCSE Bulletin*, 27(1), 287-291.
- Schmidt, K., & Bannon, L. (1992). Taking CSCW seriously. Supporting Articulation Work. *Computer Supported Cooperative Work*, 1(1-2), 7-40.
- Seaman, C. B. (1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, 25(4), 557-572.
- Simon, H. A. (1982). *The sciences of the artificial* (2 ed.). Cambridge, MA: MIT Press.
- Sommerville, I. (2007). *Software Engineering* (8 ed.). Harlow, UK: Addison-Wesley.

- The Society for Ethnomusicology. (2008). What is Ethnomusicology. Retrieved 07.05 2008, from <http://webdb.iu.edu/sem/scripts/aboutus/aboutethnomusicology/ethnomusicology.cfm>
- Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, 3(1), 36-59.
- Weinberg, G. (2002). *The Aesthetics, History, and Future Challenges of Interconnected Music Networks*. Proceedings of the International Computer Music Conference (ICMC 2002), Göteborg, Sweden.
- Weinberg, G. (2005). Interconnected Musical Networks: Toward a Theoretical Framework. *Computer Music Journal*, 29(2), 23-29.
- Zeitlin, P., & Goldberger, D. (2002). *Understanding music theory*. London: Omnibus press.

## **Appendiks A**

Dette vedlegget inneholder instruksjoner for videopresentasjon, tilgang til kildekode og veiledning for installasjon av Collaborative Music Organizer (CMO).

### **Videopresentasjon**

Gå til <http://www.student.uib.no/~bhe086/cmo/>

Videopresentasjonen viser grunnleggende funksjonalitet i CMO, og inneholder både individuelt arbeid og samarbeid. Det er laget to videoer med forskjellig oppløsning. Videoene kan også lastes ned lokalt (se nedenfor). Tid: 10 min.

For avspilling kan det brukes en nettleser hvis man har et Windows operativsystem. For avspilling i andre operativsystemer, anbefales nedlasting av VLC media player:

<http://www.videolan.org/vlc/>

Dette programmet er open source og kan benyttes i Windows, Mac OS X, BeOS, Linux, og så videre.

Direktelink for nedlasting av video:

[http://www.student.uib.no/~bhe086/cmo/cmo\\_presentation\\_1024.wmv](http://www.student.uib.no/~bhe086/cmo/cmo_presentation_1024.wmv)

### **Tilgang til kildekode**

Komplett kildekode finnes her:

[http://www.student.uib.no/~bhe086/cmo/cmo\\_source/kildekode.zip](http://www.student.uib.no/~bhe086/cmo/cmo_source/kildekode.zip)

Man trenger Visual Studio 2005 for å åpne prosjektfilen, men hver klasse kan leses i Notepad eller tilsvarende.

### **Installasjon**

Gå til [http://www.student.uib.no/~bhe086/cmo/cmo\\_install/cmo\\_install.zip](http://www.student.uib.no/~bhe086/cmo/cmo_install/cmo_install.zip) for tilgang til installasjonsfilene. Kjør Setup-filen: Denne installerer programmet til programfiler. Man kan lett avinstallere programmet fra *add/remove-programs* i kontrollpanelet. Start programmet fra startmenyen (i Windows 2000 eller XP). Logg inn med:

Brukernavn="testbruker".

Passord="master".

.NET 2.0 må være installert. (Dette medfølger Windows XP og Vista).

Hvis ikke kan det lastes ned herfra:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

Dersom man ønsker å bruke Mac OS X, Solaris, Unix eller Linux, må man først laste ned og installere Mono Project: <http://www.go-mono.com/mono-downloads/download.html> (her kan man velge ønsket operativsystem).

NB. Det anbefales å se videopresentasjonen dersom serveren er utilgjengelig (det vil si at den ikke har kontakt med internett eller er ryddet bort siden den er lokalisert på Universitet i Bergen sitt område).

## Appendiks B

Her er intervjuguiden som ble brukt under intervjuene, samt en kortfattet presentasjon av dataen som ble innhentet fra intervjuene.

### Intervjuguide

1. Din erfaring fra musikkpraksis (antall års erfaring, bakgrunn, evt. instrument)
2. Hvordan går du frem for å komponere musikk? (tar du notater? opptak? tekst på papir? Kan du beskrive kort prosessen. Blir det brukt IKT-programmer? )
  - 2.1.Hvordan oppbevarer du den komponerte musikken? (gjør du evt. blir spilt inn i studio, f.eks ved hjelp av : hukommelse, noter, tekstnotater, midi, lydopptak, diktafon, sequencer, etc.)
  - 2.2.Hva skjer med musikkideer som er uferdige? Hvordan oppbevares dette? (kan det gå tapt? Har det hendt?)
3. Hvordan overleverer du nye komposisjoner til andre bandmedlemmer? (Er IKT brukt i denne prosessen?)
  - 3.1. Hvordan kommuniserer du informasjon om komposisjonene til bandmedlemmene? (f.eks. tlf, e-post, direktemeldinger, brev, etc. Hvilke kommunikasjonsmedier er foretrukket?)
  - 3.2. Hvordan får du tilbakemeldinger om komposisjonene fra bandmedlemmene?
4. Har du noen gang samarbeidet med å komponere musikk med andre musikere?
  - 4.1.Hvordan gikk dere frem?
  - 4.2.Måtte alle bandmedlemmene være tilstede for at komponeringen kunne ta sted?
  - 4.3.Hvor (på hvilket sted) foregikk komponeringen?
  - 4.4.Var det noen som hadde hovedansvar for komponeringen?
5. Som musiker/komponist, hvilken type dokumenter er du vant å forholde deg til som er relatert til din musikkpraksis, og som er relevant for dine medmusikere? (dokumenter som f.eks: settlister, noter, tabs, lyrikk/sangtekst, kontrakter, etc.)
6. Har du erfart problemer med oppbevaring eller organisering av dette musikk materialet? (f.eks. ting har forsvunnet, blitt slettet fra datamaskin, ødelagt av virus, glemmer å lagre,etc).
7. **Scenario:** Sett at du fikk tak i et dataprogram der du kunne dele, organisere og spille av musikk som du hadde laget sammen med bandet ditt. Musikken deres kunne organiseres etter album eller konsert-settliste. Hvis du lagde en låt, så kunne du lastet den inn i programmet under ”Settliste Valle Hovin 2007”, og så ville resten av bandmedlemmene automatisk få tilgang til den (hvis de hadde brukerkonto). Etterpå kunne f.eks. bassisten høre på låten, gi deg feedback, stilt spørsmål om hvordan man spilte tema C eller lignende. Hvis dere komponerte sammen, kunne han ha forandret på låten eller lagt til bass-spor (gjennom f.eks. pro tools). I tillegg kunne dere chattet i programmet mens dere jobbet sammen.
  - 7.1.Ville dette vært interessant for deg?

7.2.Hvor (fra hvilket sted) tror du at du helst tatt i bruk dette programmet? (arbeid/skole, øvingslokale, studio, hjemme, fra andre personers hjem, annet)

## Data fra intervjuene

Syv personer ble intervjuet, og svarene deres er uttrykt med [spørsmålsnummer].[intervjuobjekt] (for eksempel 1.2 som betyr spørsmål/område 1 og intervjuobjekt 2).

## Komposisjon, bruk av IT og oppbevaring

### 1. Antall år intervjuobjektene hadde drevet med musikk og bakgrunn

- 1.1. 16 års erfaring – bass, frilans, også vært lydtekniker
- 1.2. 15 års erfaring – gitar, eget studio, produsent, synth
- 1.3. 33 års erfaring – eget studio (albumopptak, reklame), produsent, gitar, synth
- 1.4. 17 års erfaring – synth/trommer, musikkutdannelse fra høyskole
- 1.5. 14 års erfaring – gitar, komponist klassisk (for dataspill, tv-produksjoner), lærerhøgskole musikk, frilans, jobber også som musikk lærer
- 1.6. 20 års erfaring – bassist, konservatorieutdannet, frilans, driver eget studio (tv-reklame, bandopptak)
- 1.7. 13 års erfaring - Låtskriver, artist, DJ, frilans, gitarist.

### 2. Hvordan de går frem for å komponere musikk

- 2.1. Opptak av melodiideer. Strukturerer og harmoniserer det etterpå, til slutt lages det en masterblekke (type "Realbook") på det hele. (red.anm.: noter/frasering/rytmikk).
- 2.2. Notasjon, arrangering og orkestrering
- 2.3. Notepapir, korte midi-skisser, riff/akkorder innspilt i en stereo wave-fil. Forskj. skisser blir satt sammen til en komposisjon.
- 2.4. Bygge struktur (noter for å få folk til å forstå, jamming/improvisasjon).
- 2.5. Høre melodien for seg, av og til vha programvare, skrive ned akkorder, bruke trommegrooves og spille inn etter dette, først melodi, noen ganger riff.
- 2.6. Varierer: instrument gitar/piano – lage riff, melodi og akkorder – gjøre så opptak av dette.
- 2.7. Opptak av ideer på minidisc.

### 3. Har de erfaring fra IT under komposisjonsprosessen (eksempler)

- 3.1. Sibelius, audio software (Samplitude), diktafon eller mobiltelefon til opptak av melodiideer, notasjonsprogram til å lage blekke.
- 3.2. Bruker både Digidesign Pro Tools og Cakewalk Sonar
- 3.3. Win XP, Cakewalk, Steinberg Wavelab og Steinberg VStack, Digidesign Pro Tools Le, NI Kontakt 2.0, NI FM7/FM8

- 3.4. Sibelius (notasjonprogram), garage band (mac), mobiltilf (synger melodilinjer på denne) blir brukt når ikke datamaskin er tilgjengelig.
- 3.5. Sibelius, opptaksmaskin Zoom Harddiskrecorder, Cubase 4, EZ Drummer, East West Quantum Leap (stort samplearkiv med strykere, klassiske instrumenter etc.)
- 3.6. Cubase SX2.0, Cakewalk Sonar, mobiltelefon til å ta opp ideer.
- 3.7. Tidligere brukt ACID3 til å bygge låter, men bruker det lite i dag.
- 4. Hvordan de pleier å oppbevare komposisjonene sine**
  - 4.1. Noter og lydopptak
  - 4.2. MIDI, notasjon og audio opptak (wave)
  - 4.3. Midi og wave
  - 4.4. Sequencer, midi, diktafon, ark med notasjon, besifring, noe blir overlatt til hukommelsen.
  - 4.5. Mp3 og wave lagret på PC, cubase export, mapper/cd'er, noteark (blir mindre brukt nå til dags), har tidligere basert seg på hukommelse, men lært at dette er ikke lurt.
  - 4.6. skrive ned form/akkorder, opptak i studio -> bruker studio aktivt i komponering, stoler noen ganger på hukommelse - det funker av og til.
  - 4.7. tekstnotater, hukommelse, minidisc og harddisk-opptak.
- 5. Hva skjer som regel med uferdige ideer og komposisjoner.**
  - 5.1. De går som regel tapt hvis de ikke kommer forbi diktafonstadiet
  - 5.2. Oppbevarer slikt i en egen mappe med utallige ideer og skisser på pc'en.
  - 5.3. Midi/wave. Det er sjeldent ting blir glemmt så lenge man er konsekvent med å notere eller spille det inn.
  - 5.4. Ustrukturert, ofte bakerst i hodet, har hendt at ting har blitt glemmt.
  - 5.5. Har over 70 ideer liggende på opptaksmaskin, kan hende man glemmer å ta vare på prosjekt i Sibelius/Cubase.
  - 5.6. Oppbevares på HD (som musikkprosjekt/musikkfil), i arkiv hvis nedskrevet. Dette kommer an på band. (bruker f.eks. aldri noter når man spiller rock)
  - 5.7. Opptak har tidligere forsvunnet og da forsvinner idéen også. Tar vare på ideer så lenge som mulig. Har opptak som er 10 år gamle.

## Overlevering og kommunikasjon

- 1. Fremgangsmåter for overlevering av musikkmateriale til medmusikere.**
  - 1.1. Noter, hvis ønskelig audio (mp3, wave). Vha e-post.
  - 1.2. CD med preproduksjoner (mp3), midi, notasjoner. E-post er brukt.
  - 1.3. MP3-filer via E-post.
  - 1.4. Live-presentasjon og forklaring, spille inn på garageband m/bass trommer -> spille av i øvingslokalet. Leverer fysisk eller sende e-post. (hadde ikke fast internett-tilgang. Brukte mobiltelefon).

- 1.5. Sender MP3 via gmail, har egen webside med repertoar av MP3 (som tilsvarer feks. en settliste som skal fremføres LIVE), brenne CD når man skal treffe folk ansikt til ansikt.
- 1.6. Vise med instrument i øvingslokalet. Sender filer via e-post (mp3).
- 1.7. Viser skisser, brenner opptak fra minidisk til CD, eller sender via e-post.
- 2. IT-programmer som blir brukt til dette (alle bruker IT)**
  - 2.1. Sibelius, eventuelt samplitude
  - 2.2. Bruker egen server i studiosammenheng med internasjonale musikere, (laster opp/ned wavefiler som de recorder i eget studio), FTPClient (freeware), E-post,
  - 2.3. Wavelab, e-post
  - 2.4. Sibelius, garageband. Mye går ut på å vise de andre mens en selv er tilstede.
  - 2.5. E-post, brenneprogram, internettleser,
  - 2.6. Trommesequencer, media player (konvertere til mp3)
  - 2.7. Bruker PC'en til å konvertere filer og lagre ideer, samt sende e-post.
- 3. Hvordan de kommuniserer med sine medmusikere om komposisjonene**
  - 3.1. tlf, mail og personlig oppmøte
  - 3.2. Msn, e-post, og dels tlf.
  - 3.3. E-post
  - 3.4. garageband, vise til bandmedlemmer, nynne stemmer, samspill
  - 3.5. Tlf eller e-post.
  - 3.6. tlf, ansikt til ansikt. Noe e-post.
  - 3.7. Hovedsaklig e-post.
- 4. Hvordan de får tilbakemeldinger om komposisjonene fra sine medmusikere**
  - 4.1. Tlf, mail og personlig oppmøte
  - 4.2. Msn, e-post, og dels tlf.
  - 4.3. E-post, telefon
  - 4.4. Muntlig, tlf, ansikt til ansikt
  - 4.5. Tlf eller e-post
  - 4.6. Låter som ble sendt via e-post -> får tilbake kommentarer på e-post. Ellers snakkes man i øvingslokalet eller når man generelt treffes.
  - 4.7. Muntlig, via telefon, sms, e-post, brev og nye opptak basert på komposisjonen hans.
- 5. Kommunikasjonsform som foretrekkes**
  - 5.1. Personlig oppmøte
  - 5.2. Msn, e-post, og dels tlf. (ingen preferanser)
  - 5.3. E-post
  - 5.4. Muntlig.
  - 5.5. tlf, sms, bluetooth med gitarelever (overføre tabulatur).
  - 5.6. Ansikt til ansikt
  - 5.7. Utveksling av minidisc. (men synes også at mp3-filer via e-post er mest økonomisk).



## Hvordan fungerer samarbeidende komposisjon

### 1. Hvorvidt de har samarbeidet med andre musikere om komposisjon av musikk (alle har erfaring fra slikt samarbeid)

- 1.1. ja
- 1.2. Foretrekker å jobbe selvstendig, men har gjort det tidligere.
- 1.3. ja
- 1.4. ja
- 1.5. ja, men ikke så mye nå lengre
- 1.6. ja
- 1.7. ja

### 2. Fremgangsmåte for samarbeidende komposisjon

- 2.1. Lage riff, bruddstykker, brainstorm
- 2.2. Både improvisasjon og forhåndsarbeidete ideer
- 2.3. Improvisasjon og utbygging av riff/bruddstykker
- 2.4. Forskjellig: jamming/improvisasjon, tidligere komponert tema som blir utbrodert sammen, laget en grunnpresentasjon -> og så fikk de forskj bandmedlemmene ansvar for egen stemme.
- 2.5. Folk kom med ideer (riff, bruddstykker), utviklet sammen. Men ofte ble utformingen av en gitt stemme gjort av den som kunne spille det aktuelle instrumentet/stemmen.
- 2.6. Ideer, innspill fra andre – utviklet mot en ferdig låt. Evt. låten var klar men arrangere den sammen. Noe improvisasjon.
- 2.7. Improviserte frem ideer som ble arrangert i ettertid. Noen ganger basert på enkle riff.

### 3. Om de har erfart/finner det nødvendig at alle bandmedlemmer er tilstedeværende under samarbeidende komposisjon

- 3.1. Nei
- 3.2. Nei
- 3.3. Nei
- 3.4. Nei, det holder med noen få. Men dette kan få problemer senere hvis ikke alle føler de får være med, + at kreativiteten kan begrenses når man får ansvar for en stemme som man egentlig ikke trakterer godt.
- 3.5. Nei
- 3.6. Nei (f.eks, bass og trommer kunne vært nok for å lage komp)
- 3.7. Nei, men foretrekker at de fleste er der.

### 4. Sted der samarbeidende komposisjon har funnet sted

- 4.1. Øvingslokalet og hjemme
- 4.2. Øvingslokalet
- 4.3. Studio
- 4.4. Øvingslokalet, en sjelden gang hjemme med andre.
- 4.5. Øvingslokalet, rockeverkstad, hjemme/borte hos andre musikere.

4.6. Øvingslokalet.

4.7. Øvingslokalet.

**5. Hvorvidt de mente at noen pleide å ha hovedansvar for komponeringen**

5.1. Ja

5.2. Ja

5.3. Ja

5.4. I noen bandsettinger er lederfiguren tydelig, andre bandsettinger mindre tydelig.

5.5. Muligens, men ikke fast inndelt ansvar.

5.6. Variert fra band til band. Intervjuobjekt har vært hovedansvarlig ved flere anledninger.

5.7. Som oftest gitarist og bassist.

## Oppbevaring og organisering av dokumenter

**1. Hvilke typer dokumenter de er vant til å forholde seg til**

1.1. Settlistre, noter, sangtekst, kontrakt, raider, kjøreplan.

1.2. Settlistre, notasjon

1.3. Noter, kontrakter, sangtekst

1.4. Ensemblekart, notasjon (m besifring), tekst, settlistre, utkast på papir.

1.5. Noter, tabulatur, settlistre,

1.6. Settlistre, blekker, akkordskjema. Noter på viktige detaljer.

1.7. Settlistre, sangtekster og kontrakter. Også regnskap (excel).

**2. Problemer med oppbevaring eller organisering av musikkmateriale**

2.1. Mobiltelefonkrasj, diktafonnotater forsvunnet

2.2. Nei, heldigvis

2.3. Harddiskfeil og sammenbrudd har gjort at ting har forsvunnet

2.4. Brukerfeil = glemmer å lagre ting i et lydopptak-program/sequencer, Virus: sletting av materiale, av og til har cd/opptak forsvunnet.

2.5. En CD har forsvunnet. Tidligere basert på hukommelse; kunne være problem hvis man glemte hvordan låten var satt sammen, PC: gjort feil/skrevet over annet materiale, man tror man har eksportert, og så har man klikket "save->no", nummerering av prosjekter gått galt, CD blitt ødelagt,

2.6. Nei

2.7. Tekster, settlistre og utskrifter har en tendens til å bli borte på konserter og øvinger, og datamaskiner har blitt infisert med virus.

**3. Utdyping av scenario som beskriver programmet som utvikles i masterstudiet. Hvorvidt dette høres interessant ut for dem som komponister?**

3.1. Ja

3.2. Ja, det kunne vært en ideell løsning å presentert komposisjoner lettere.

3.3. Konseptet høres interessant ut

3.4. Ja, høres ut som man kan spare tid, enklere å henvise til låter og et gitt parti i låten, mulighet for felles brainstorming online uten at man er på samme sted.

3.5. ja

3.6. ja, låtene blir lett tilgjengelig for alle.

3.7. ja, ikke minst fordi man trenger ikke å være avhengig av eksterne chatte og mail-programmer. Tagging av forskjellige sekvenser i audiofilen og notearket med tekst (kommentarer, forslag og spørsmål) ville også vært interessant.

**4. Hvor de tror at de vil ta i bruk et slikt program (hypotetisk spørsmål)**

4.1. Hjemme

4.2. Hjemme

4.3. Studio og hjemme

4.4. Hjemme, kanskje fra skolen (musikkutdanning).

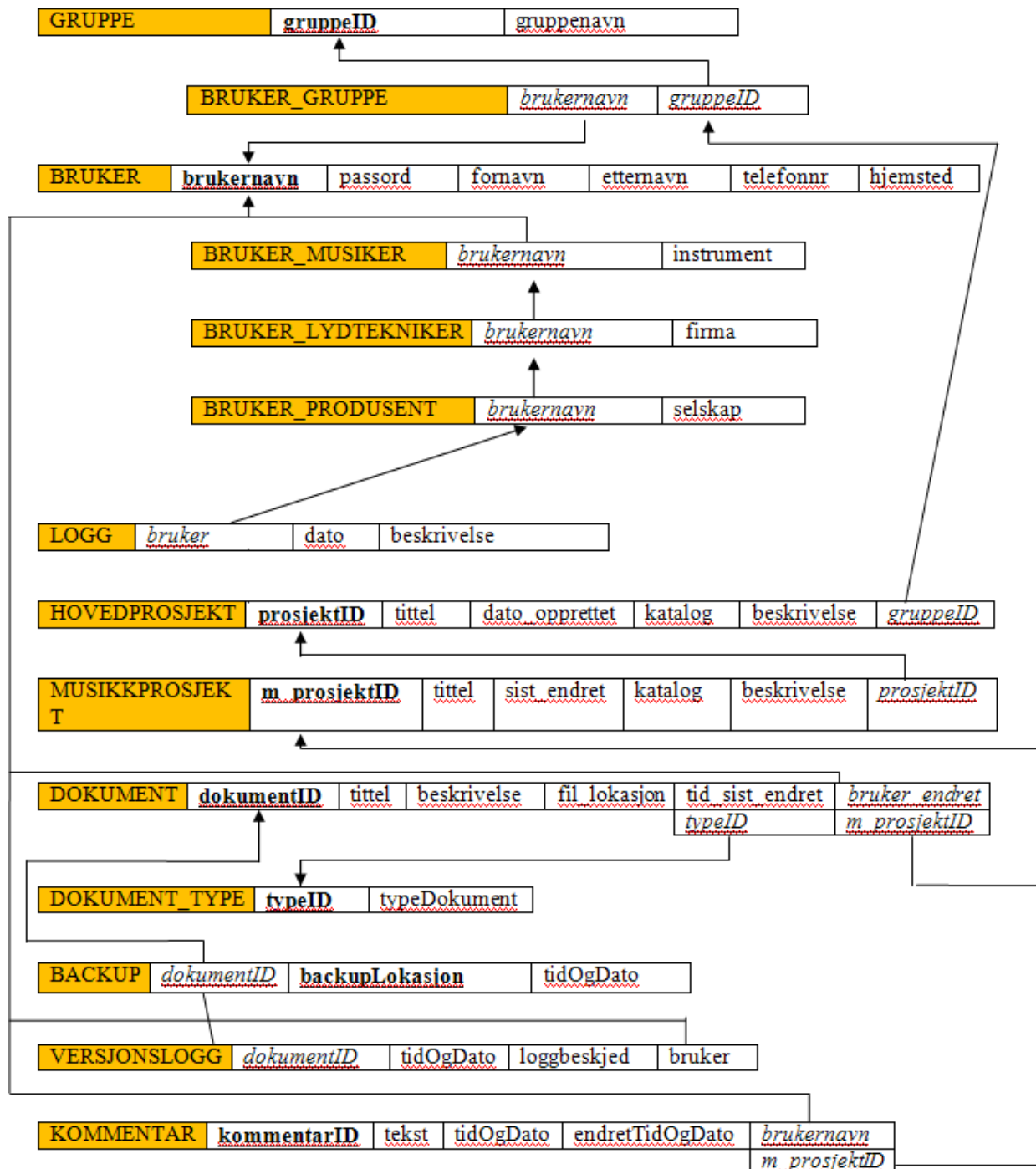
4.5. Hjemme, fra jobb (musikklærer/kultur), kanskje på toget/buss med trådløst nett.

4.6. Hjemme og studio

4.7. Hjemme, men også fra skole/studio. Gjerne fra privat PC når man er på ferie eller reise (i forbindelse med turnevirksomhet).

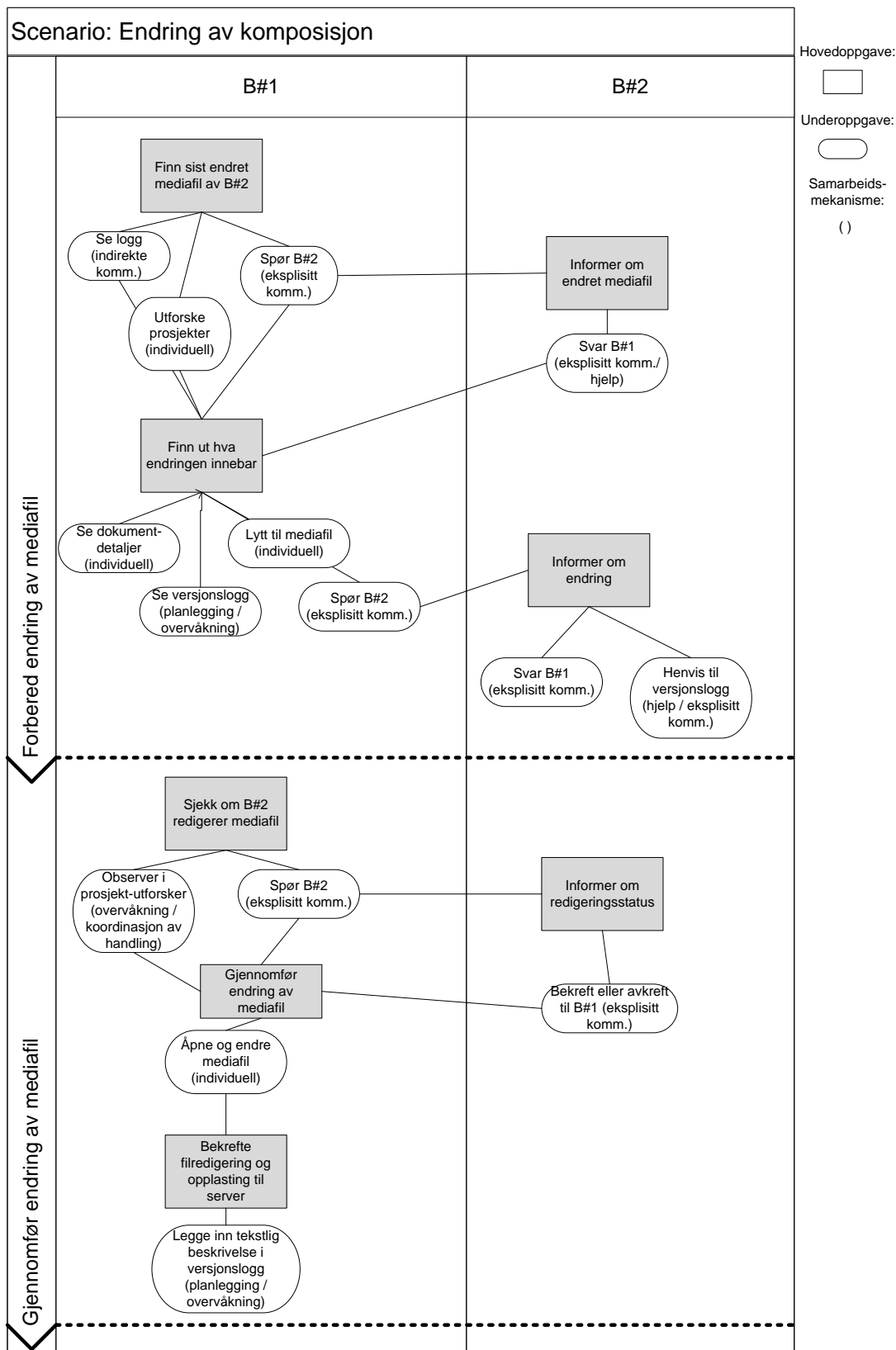
## Appendiks C

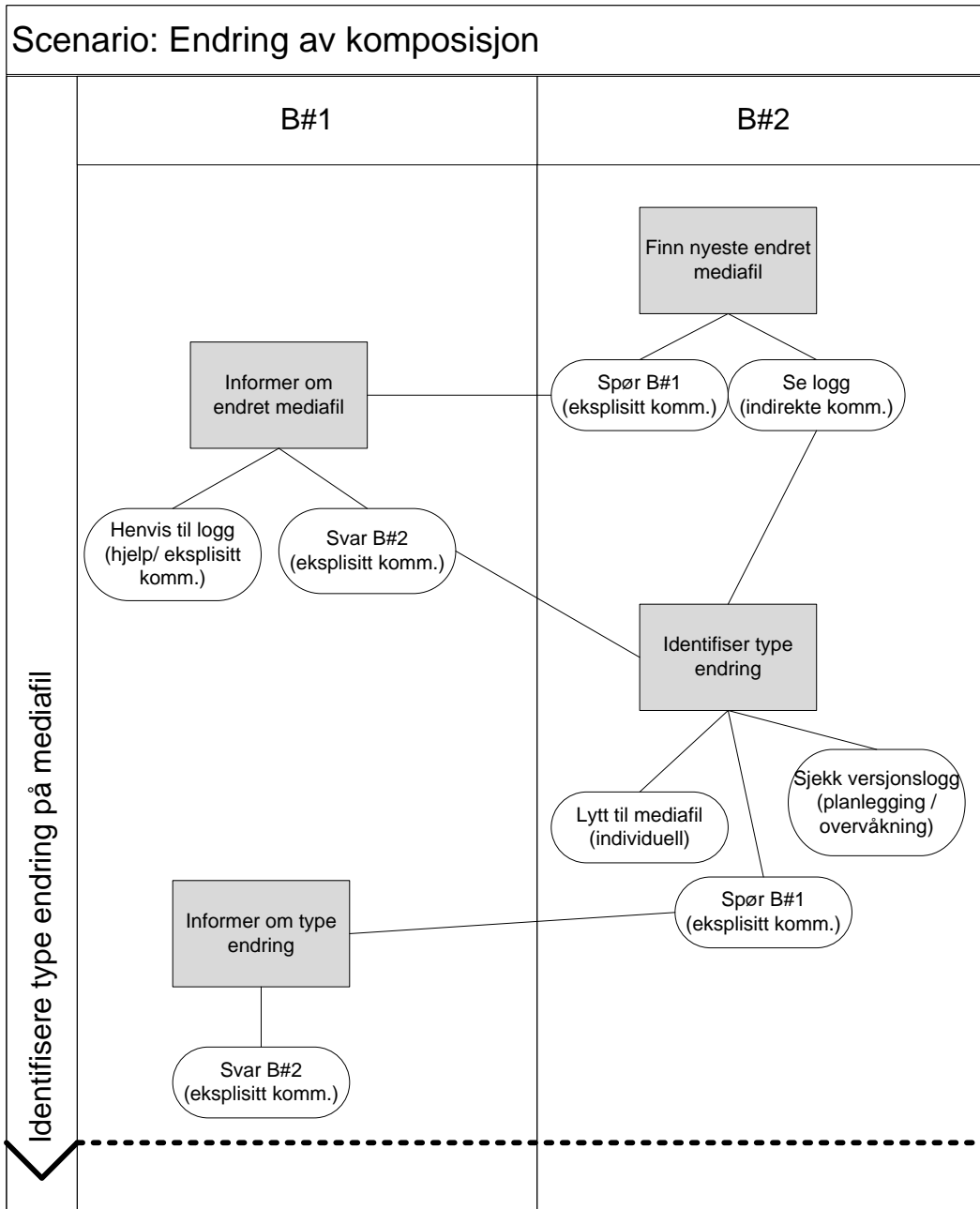
Relasjonsmodell. Basert på EER-modellen. Viser oppbygningen av relasjonsdatabasen.

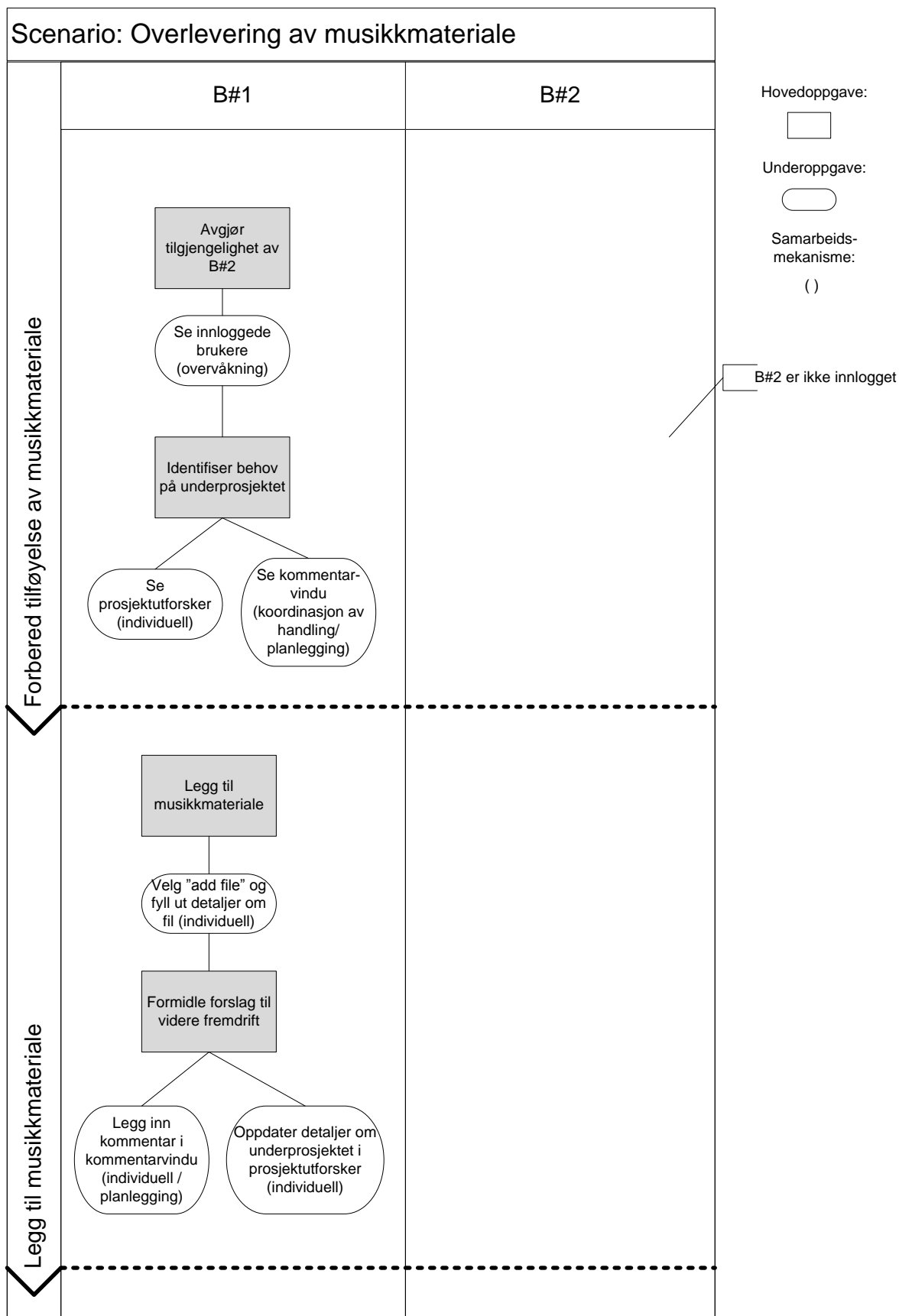


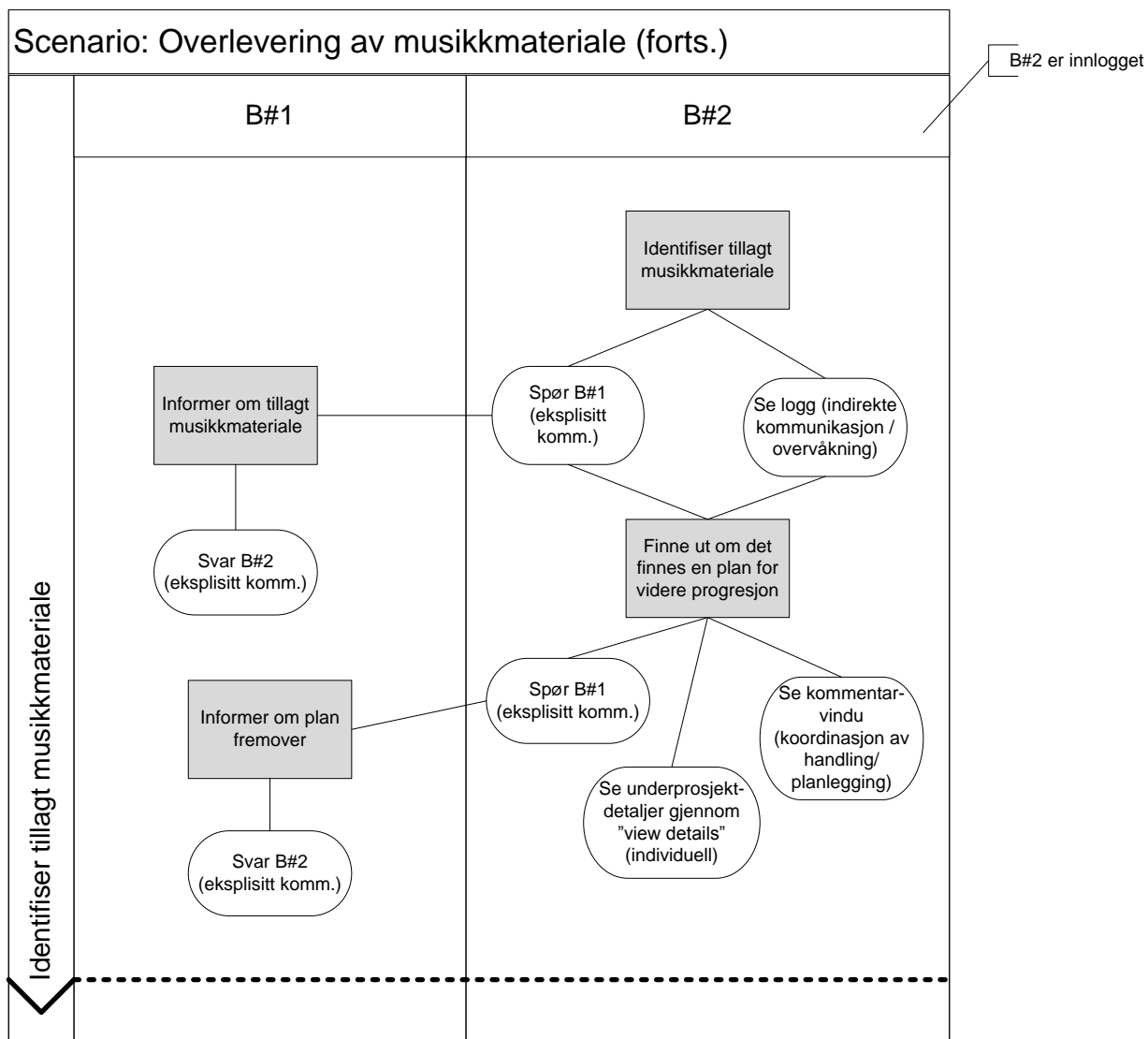
# Appendiks

D

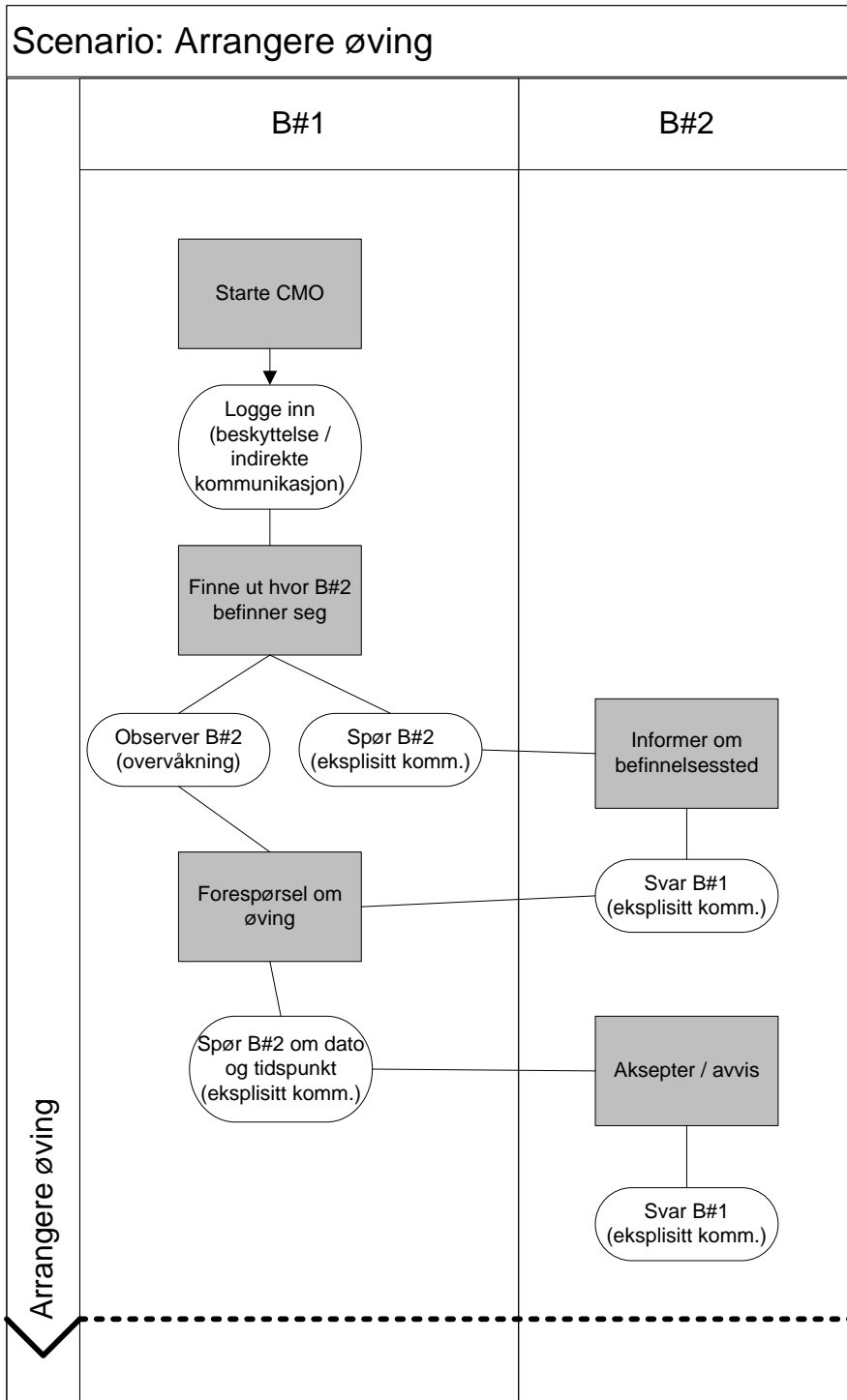












Hovedoppgave:



Underoppgave:



Samarbeids-  
mekanisme:

()

## Appendiks E

Her vises oppgavene som ble gjennomgått med brukerne i evalueringen av CMO. Oppgavene er basert på analysediagrammenes oppgaver under bruker B#1. Undertegnede hadde rollen som B#2 i alle scenarioene.

### Scenario: Diskusjon om komposisjon

1. Finn ut om B#2 er tilgjengelig
2. Spør B#2 om han har mulighet til å ta en liten diskusjon.
  - a. Vent på svar.
3. Fortell at du ønsker å diskutere underprosjektet «Over the wall»
4. Diskuter
  - a. Påpek svakheter/styrker ved underprosjektet
  - b. Spør B#2 hva han mener.

### Scenario: Endring av komposisjon

Du skal i denne oppgaven endre/redigere en komposisjon (wave-fil) som er lagt inn i systemet.

1. Finn sist endret mediafil av B#2
2. Finn ut hva B#2 gjorde i løpet av sin siste endring.
3. Sjekk om B#2 også redigerer den filen du skal redigere.
4. Gjennomfør en endring på filen.
  - a. Eksempel: Åpne filen først (gjennom «Open With») og velg Wavelab som ønsket program.
  - b. Når filen er åpnet, velg «Normalize» i wavelab, og lagre filen.
5. Bekreft filredigering og opplasting til server.

Du er nå ferdig med dine enkeltstående oppgaver. Hvis B#2 lurar på hva du har gjort (gjennom chat-vinduet), må du informere ham om dine handlinger.

## Scenario: Overlevering av musikkmateriale

Du skal i denne oppgaven legge til en komposisjon i systemet slik at den blir tilgjengelig for B#2.

1. Finn ut om B#2 er tilgjengelig i systemet
  - a. Hvis han er tilgjengelig, spør ham om hva som har blitt gjort og hva som bør gjøres på underprosjektet «Over the wall» som ligger i hovedprosjektet «The legacy».
2. Hvis B#2 ikke er tilgjengelig, prøv å identifiser hva som har blitt gjort tidligere på «Over the wall» og om B#2 har skrevet hva som bør gjøres. (Dette for å forsikre at du ikke legger til et musikkbidrag som allerede har blitt lagt til tidligere).
3. Basert på det du har funnet ut, kan du velge mellom å legge til 3 forskjellige musikksnutter (alle snuttene finnes i «Mine Dokumenter\Scenario2»):
  - a. «Rytmask grovskisse»
  - b. «Vokalstemme»
  - c. «Gitarsolo»
4. Legg til snutten på underprosjektet «Over the wall».
5. Legg til informasjon om forslag til videre fremdrift på låten.
  - a. Her kan du eksempelvis skrive at du foreslår at hele bandet samler seg og øver inn låten slik at den kan brukes på neste konsert.
  - b. Eller at bandets lydtekniker mikser låten slik at den kan videresendes til et plateselskap.
6. Du er nå ferdig med dine enkeltstående oppgaver. Nå vil B#2 logge seg inn i systemet, og hvis han spør om hva du har gjort, så må du svare ham på spørsmål som er relatert til arbeidet du nettopp har utført.

## Scenario: Arrangere øving

1. Logg inn i Collaborative music organizer.
2. Finn ut hvor B#2 fysisk befinner seg.
  - a) Hvor er hjemstedet til B#2?
3. Dersom B#2 befinner seg innenfor ditt geografiske område: spør ham om han har mulighet til å være med på en øving den 8. april

## Appendiks F

Her vises et skjermbilde av serverprogrammet som håndterer tilkoblingene mellom klienter, sending av meldinger, status, og så videre.

